



Fall 2019-2020

Home

Announcements

Assignments

Piazza

Grades

Pages

Files



Calendar



Inbox



Help

10. Dialogflow

Due Dec 13, 2019 by 11:59pm **Points** 10 Submitting a text entry box or a file upload **File Types** zip and js
Available Nov 26, 2019 at 8:30pm - Dec 20, 2019 at 11:59pm 24 days

This assignment was locked Dec 20, 2019 at 11:59pm.

Homework: Dialogflow (10 Points)

This assignment is meant to teach you the basics of building a dialog-based interface. You will do this by working with Dialogflow, a software suite by Google, as well as a react app that is provided to you. You are expected to create a completely functional voice agent that also modifies the application state on the server, such that the web page is updated.

You can clone the [WiscShop](#) repository to your machine, and run it with `npm install` and `npm start`. You will be creating a Dialogflow agent from scratch, and uploading the zip files for the agent, as well as your `webhook.js` file from the repository.

Potential Gotchas/Helpful Suggestions:

- We highly suggest reviewing the rubric and specification below. Information on using the API can be found in the [WiscShop](#) project README.
- Feel free to peruse the [WiscShop](#) code. It has a number of fetch examples with more sophisticated filtering of products.
- See [Getting Started](#) for details on getting the webhook running.
- If you want to see how to update the application without dialogflow, feel free to use [Postman](#) to do the API calls.

Submission Details:

To submit the assignment, you will need to do **two** things.

- The first thing you will be providing is the zip file of your dialogflow agent. You can download this file from the settings pane:

- Secondly, you will be providing the `webhook.js` file, a starter of which is provided in the [WiscShop](#) repository.

Specification:

- Login**
 - User is able to login with username and password. You do not need to handle account creation.
 - NOTE:** It is sufficient if the user enters this information as a text query (typing), in case the username and/or password is hard to parse. It should still be english, e.g. "Log in with username <username> and password <password>".
- Queries**
 - Categories:** User should be able to query about the types of products offered.
 - Tags:** User should be able to inquire about the types of tags for a specific category.
 - Cart:** User should be able to request information about what is in their cart (e.g. total number and type of items, total cost, etc.).
 - Product Info:** User should be able to request information about a product. If the product has reviews, they should be able to inquire about reviews and average ratings.
- Actions**
 - Tags:** User should be able to narrow down the search results within a category by specifying tags, e.g. "Show me all the red ones".
 - Cart:** User should be able to add/remove items (or multiple of an item) to/from your cart. They should also be able to clear their cart.
 - Cart Confirm:** User should be able to review, then confirm their cart.
- Navigation**
 - User should be able to navigate through the application with the voice assistant using natural language, e.g., "Take me to the home page" or "Show me the hats".
 - For a full breakdown of the various routes in the application, see the [WiscShop](#) readme.
- Messages**
 - Messages should be updated when the user or agent says something.
 - Upon starting a new session, the messages should be cleared.
- Design and Personality**
 - Users should be able to converse with the agent in natural language to perform tasks.
 - You shouldn't need to specify product id's to navigate.
 - The agent should have a personality, and should engage in expected turn-taking behavior.

Getting Started:

- When you clone the WiscShop Repository, you may want to set the remote to a different location, if you want to keep it under version control. **Do not commit to the WiscShop Repository.**
- After cloning, you will need to run it with `npm install`.
- You will then need to set up the webhook. You can find this in `/webhook/webhook.js`.
- To run the webhook, you will need to run two functions simultaneously, so boot up two more terminal windows in that directory.
- First, run `npm run tunnel`. You will then see a screen that looks like this:

```
ngrok by @inconshreveable                                         (Ctrl+C to quit)
Session Status: online                                         Session Expires: 7 hours, 59 minutes
```

Submission

✓ Submitted!

Dec 14, 2019 at 11:55pm (late)

Submission Details

[Download cs639_final \(1\).zip](#)

[Download webhook.js](#)

Grade: 7.3 (10 pts possible)

Graded Anonymously: no

[View Rubric Evaluation](#)

Comments:

Greetings, I have already talked to the prof. to ask for one extra late day to finish this project. zhaoyin

ZHAOYIN LI, Dec 14, 2019 at 11:55pm

```

Update          update available (version 2.3.35, Ctrl-U to update)
Version        2.2.8
Region         United States (us)
Web Interface http://127.0.0.1:4040
Forwarding    http://9ca77a7b.ngrok.io -> localhost:8080
Forwarding    https://9ca77a7b.ngrok.io -> localhost:8080

Connections   ttl     opn      rt1      rt5      p50      p90
              0       0       0.00    0.00    0.00    0.00

```

6. In dialogflow, you will want to go to the fulfillment tab, and toggle on the enable at the "Webhook" section. Replace the URL with the one provided in the tunnel, e.g.:

Webhook

ENABLED

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

URL* <https://9ca77a7b.ngrok.io>

BASIC AUTH Enter username Enter password

HEADERS Enter key Enter value

7. **NOTE:** The tunnel URL will change each time you restart your tunnel. You will have to update the URL in your dialogflow agent whenever you do this.

8. To view output of the tunnel, go to your third terminal, and type `npm run dev`. This will output logs.

9. You are now set to configure your intents with the webhook. You can see one "welcome" example in the [webhook.js](#) file.

10. To configure intents to work with the webhook, you will need to enable the webhook in that intent:

Fulfillment

Enable webhook call for this intent

Enable webhook call for slot filling

Dialogflow				
Criteria		Ratings		Pts
Login	User is able to login with the voice assistant.	0.5 pts Full Marks	0.0 pts No Marks	0.5 pts
Queries: Categories	User is able to request information about categories.	0.5 pts Full Marks	0.0 pts No Marks	0.5 pts
Queries: Tags	User is able to request information about tags for a specific category.	0.5 pts Full Marks	0.0 pts No Marks	0.5 pts
Queries: Cart	User is able to request information about their cart (e.g. total number/type of items, total, etc).	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Queries: Product Info	User is able to request information about a product. If the product has reviews, they should be able to inquire about reviews and average ratings.	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Actions: Tags	User should be able to narrow down the number of items shown based on tags.	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Actions: Cart	User should be able add/remove items (or multiple of an item) to/from their cart.	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Actions: Cart Confirm	Users should be able to review, then confirm their cart.	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Navigation: Pages	Users are able to navigate throughout the application with the voice assistant using natural language, e.g. "Take me to the home page" or "Show me the hats".	1.0 pts Full Marks	0.0 pts No Marks	1.0 pts
Messages	Messages should be updated when the user or agent says something. Upon starting a new session, the messages should be cleared.	0.5 pts Full Marks	0.0 pts No Marks	0.5 pts
Design and Personality	Users should be able to converse with the agent in natural language to perform tasks. In other words, you shouldn't have to specify product id's.	2.0 pts Full Marks	0.0 pts No Marks	2.0 pts
Total Points: 10.0				

