WIT.AI Web App

FACEBOOK Developer circles community challenge

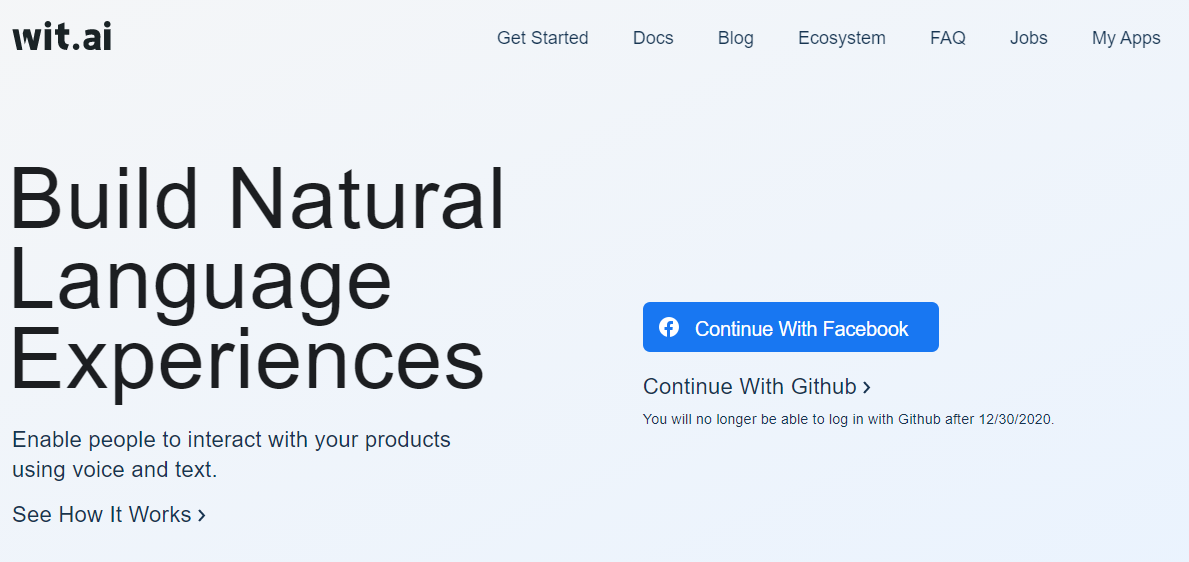
Lopez, Darian

2020

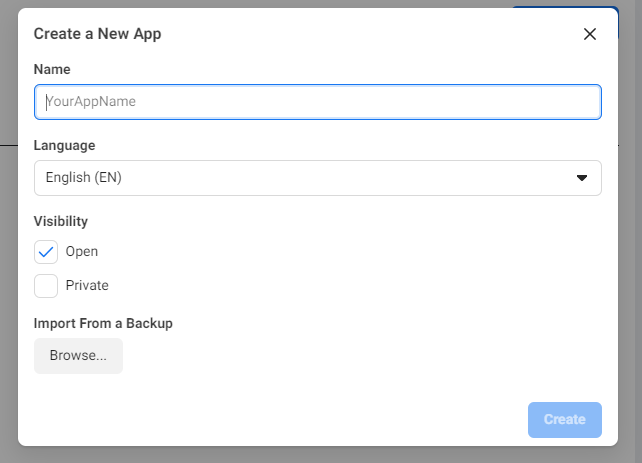
Introduction

This tutorial demonstrates the basic steps of utilizing Wit.ai to interact with basic user text input, execute corresponding function and return the output to the user. The layout of the entire steps includes but are not limited to, “Signup for Wit.ai with Facebook, Create Wit.ai application, Train the app, Improve app detection, Query application for results, and Continue to Improve application (Wit.ai, n.d).

**Step 1 Signup with Facebook**

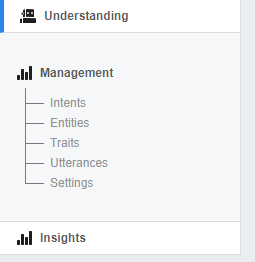


**Step 2 Create a new application**

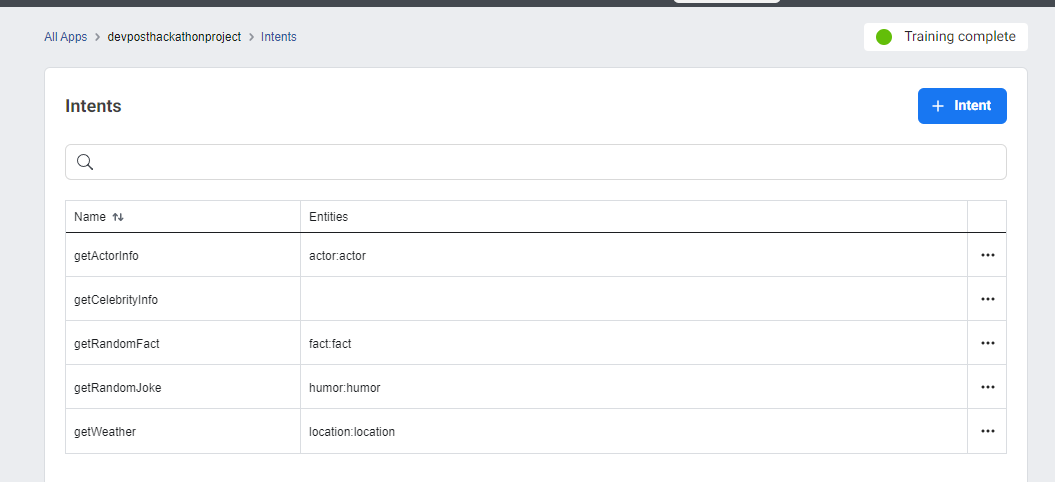


**Step 3 Begin Training your application**

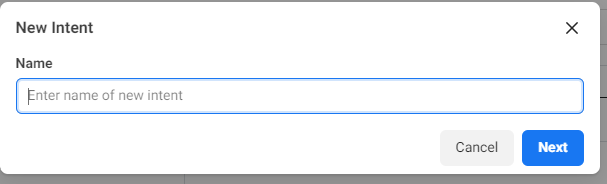
This is the portion of the creating where a large potential of your work may occur. I recommend and Wit.ai recommends starting with the Intents to start creating your project. Navigate to the Management tab on the left-hand side of the page. Then, select Intents to begin creating an Intents.



From there, you will be brought to the intents page where you can see already created intents as well as create a new one.



Select the Blue intent button shown in the above screen shot to create an Intent. Give it a name and continue.

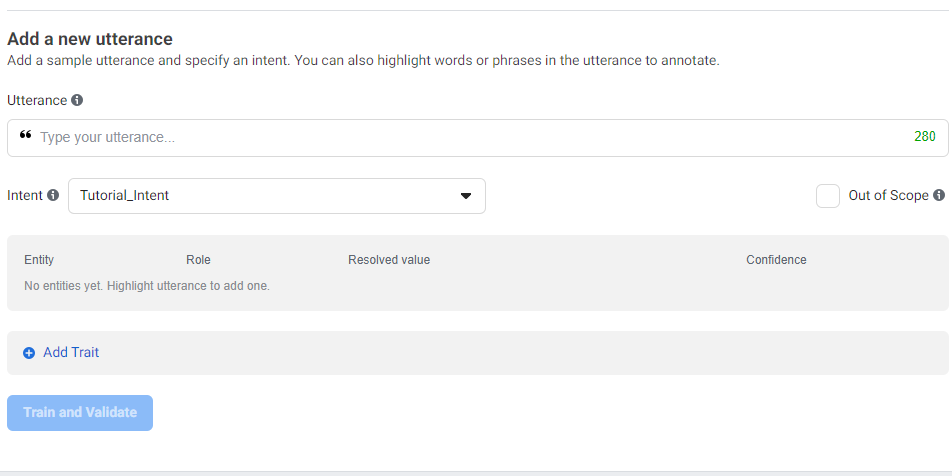


Now that we have our new Intent made, the next step to give it some training data. Navigate again to the left-hand side menu and select “Understanding”. From here you will be providing sample data and for the application to learn from. Find the section labeled “Add a new utterance”. In the field with the place holder “Type your utterance”, type in the type of input you anticipate your application will receive. For this example, we will use a few phrases:

* Is this a tutorial for WitAI?
* What is WitAI?
* Help me with WitAI?

After entering the phrase into the input field, select the “Intent” dropdown bar and select the intent you are entering the test data for. Then, select the “Train and Validate’ button below it. You ensure the intent and phrase and entered for each time you wish to train the application through adding a new utterance.

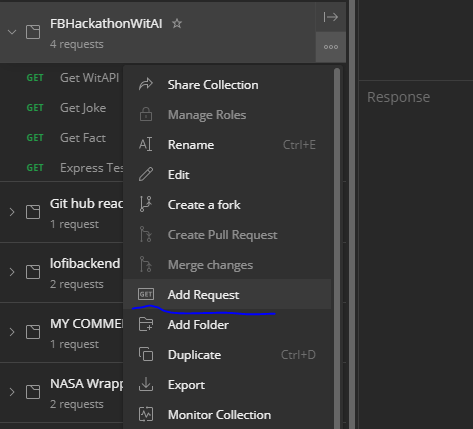
Additionally, you can improve application detection through adding more utterances and clicking the “Train and Validate” Button as mentioned previously. “The more utterances you validate, the better Wit will understand. The confidence level should start improving. (Wit.ai, n.d.).”

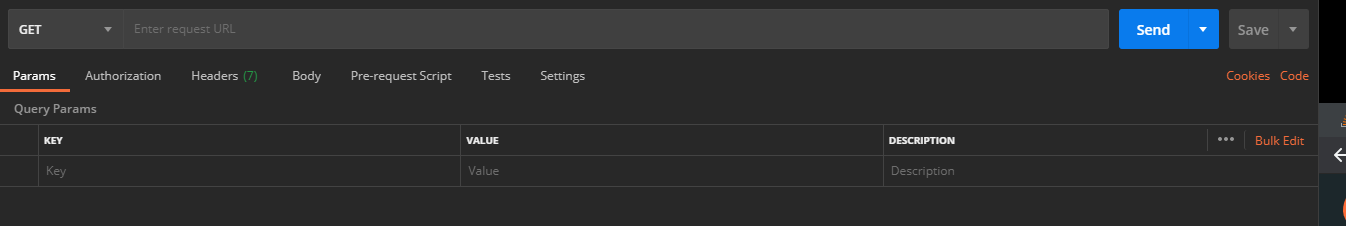


**Step 4 Query the application to see your results**

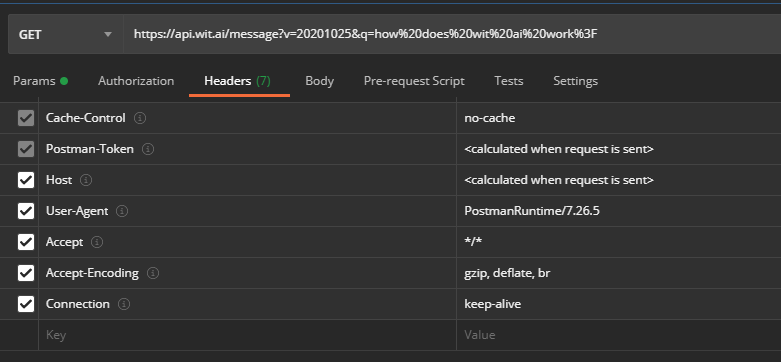
The next step should be enjoyable as you should start to see how your Wit.ai application can perform. For this step we will navigate to Settings and locate the HTTP API field. This field allows you to enter an utterance like before but in real time to query and see the response from your Wit.ai application! This field will generate a curl command you could run through the command line. However, in this example we will utilize Postman. If you don’t already have postman installed you can get the program at this link <https://www.postman.com/downloads/>.

Once you’ve downloaded the application, open it to get started. With the program open, select new Collection and create a collection to store your requests in. The requests you can generate with Postman allow you to send network traffic just as if you were using your internet browser. You can load websites, receive website data, errors, files and much more. With said, once you’ve created your collection, create a new Request. With your new request open, in your newly created collection, you should see the similar visuals in the following screenshots

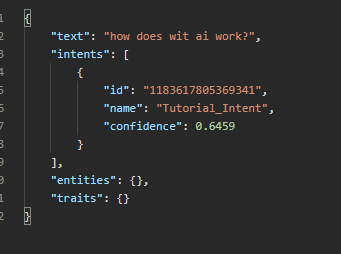




The next step is to copy the HTTP URL provided in the HTTP AI curl output. For me the URL was <https://api.wit.ai/message?v=20201025&q=how%20does%20wit%20ai%20work%3F>. Copy the URL into post man and ensure it is on GET, not POST, DELETE, or PUT. Now before we send out request to our all mighty AI, we need the authorization token. Under the URL field, locate the Headers Tab. It will look something like this:



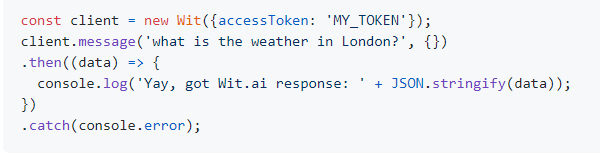
You are going to click the key empty field at the bottom which the placeholder “key” and Enter “*Authorization”*. In the corresponding empty field to the right with “Value” it in, you are going to want to enter “BEARER <your Server Access Token>” just like that. Once you have them filled out, you can submit your request wit the “Send” button and observe the output. The output should look something like this:



There you have it! You can now send text to your Wit.AI application and it will tell you what it thinks the intent is.

For future integration, you can implement this into a network chat function. A functionality example would be using NodeJS or ReactJS to interact with the user, send the users, request to Wit.ai, get the corresponding intent, and query the related API call or function.

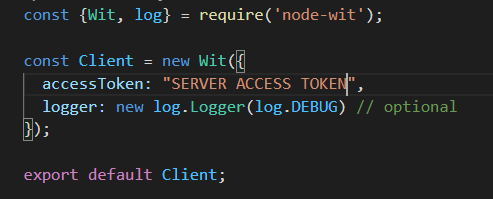
This next tutorial section is built on ReactJS and uses the Wit.ai module as provided on Wit.ai’s documentation page. Their GitHub page here - <https://github.com/wit-ai/node-wit> provides the basics for getting the skeleton of a Wit.ai client created.



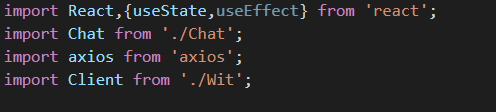
For this next section, partial code snippets will be captured but not every line of code will be provided in this documents. The entirety of this project can be found on this Hackathon submission’s GitHub repository. For structure, we will be creating the following components alongside the entry point App.js file : Wit, Chat, Chat Window, and Header.

**Wit.js**

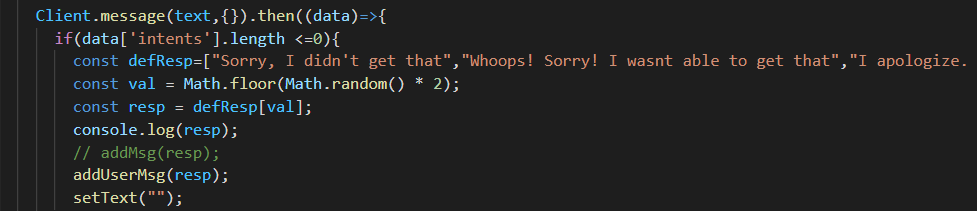
The Wit.js component will be like the screen capture above and will be the client from which all Wit.ai application queries are made.



Next, we navigate to the ChatWindow component to integrate the Wit.ai client with the ReactJS application. We will use the React Webhooks useState, and axios to make web calls and handle the state of the application.

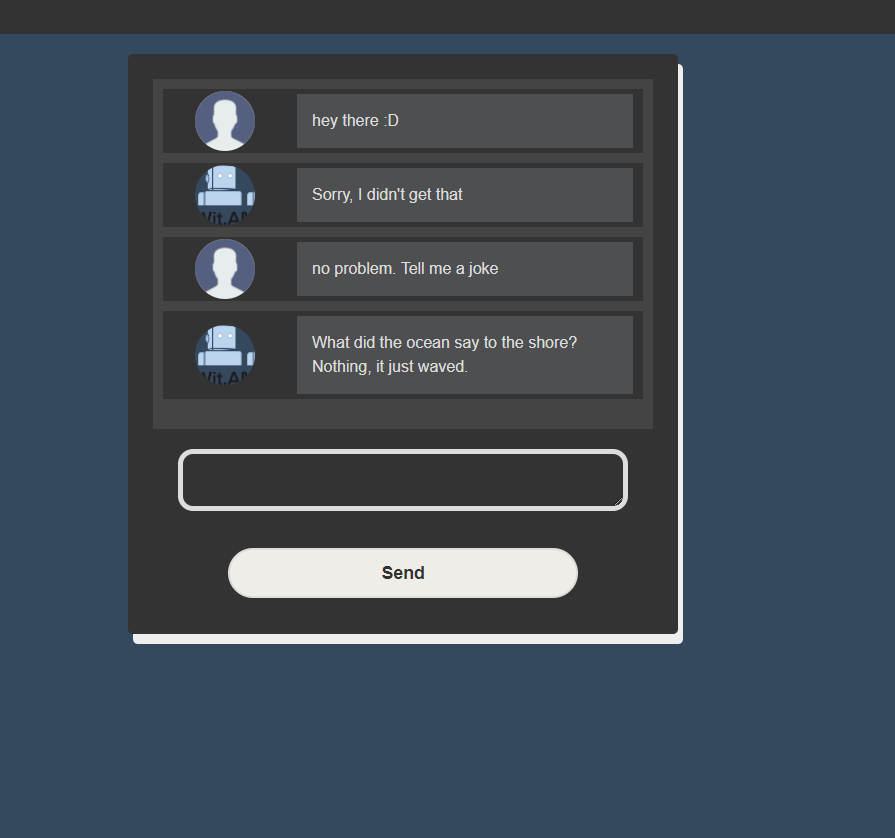


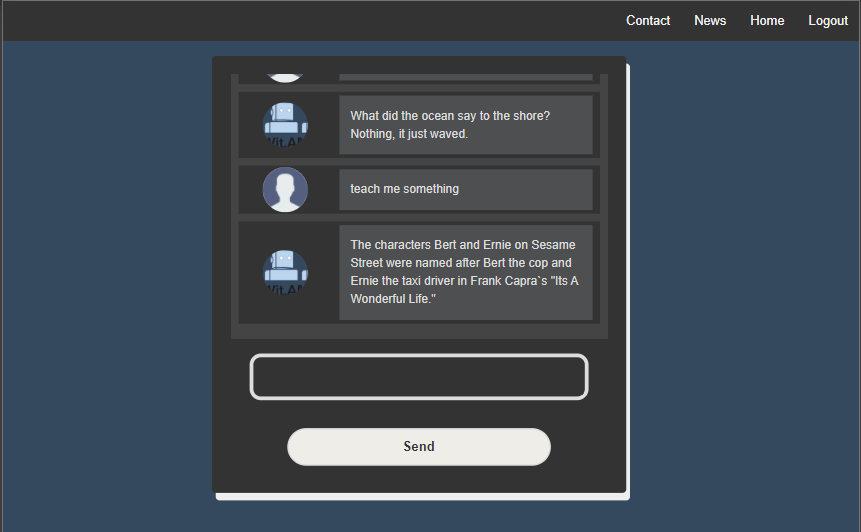
The key part of this component is the Wit.ai client processing the text send trough the users text field and the corresponding Axios network call thereafter. The corresponding capture follows the Client receiving the user input, processing it, getting the intent, and making a response or api call based off the Wit.AI determination.



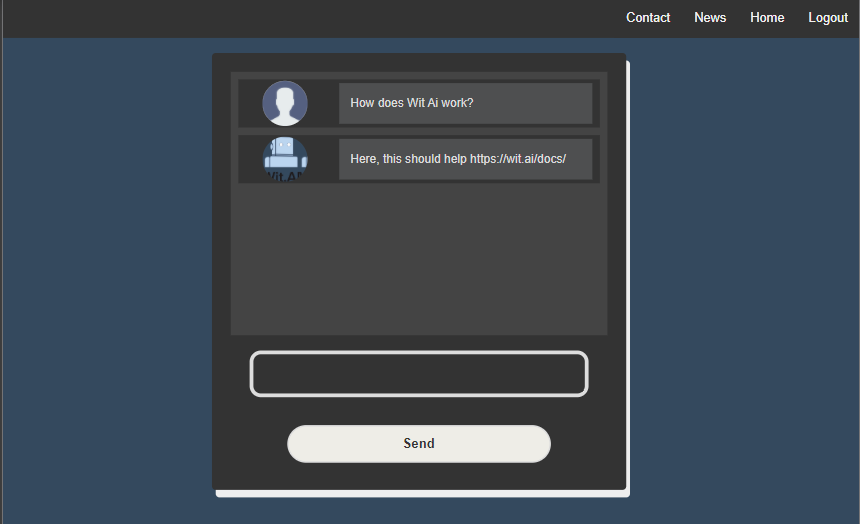


When applied to some front-end UI construction, a simple implementation will look like this (Note, the final example is based on the material covered in this walkthrough):





**Our tutorial example implemented!**



**Resources**

Wit.ai. (n.d.). Build Your First Wit App. Retrieved October 25, 2020, from https://wit.ai/docs/quickstart

Wit-Ai. (n.d.). Wit-ai/node-wit. Retrieved October 25, 2020, from https://github.com/wit-ai/node-wit