

Spot Market Chatbot

This is our repository for spot market chat bot implementation. You can work with the chatbot live on Skype, Telegram and Facebook Messenger. On Telegram, simply search for @spotmarket_bot username. On Facebook Messenger and Skype, search for "Spot Market Chatbot". after finding the chatbot, you can start interacting right away.

Deployment Instructions

If you are interested in testing out the chatbot locally, or using your own machine, these instructions guide the set up of the bot and its dependencies.

Dummy API

1. Install NodeJS and NPM
2. Navigate to the api root folder
3. Run 'npm install' to install the required dependencies
4. Install MongoDB and create a collection called 'spotmarket' without any authentication requirements
5. If https is being used, add certificate as cert.pem and key as key.pm under certificates in the root api folder and set debug to 'false' in server.js
6. You can add some dummy data to the api by using mongoimport (which may need to be installed) and run 'mongoimport --jsonArray -d spotmarket --collection COLLECTION COLLECTION.txt' where COLLECTION is replaced by the name of one of the data files without the file type, the most important ones to do this for are products and retailers.
7. Run the server by running 'node server.js', note that on debian and debian based distributions such as ubuntu this command may be 'nodejs server.js' to avoid conflicting with another package.

LUIS Instance

There are two instances of LUIS under the luis repo, one is a simpler version with only two entities (Spot Market Chatbot.json), and the other is a more complicated one that follows the IAB categories, the same categories the spot market engine would follow in the future. (Spot Market Chatbot - IAB.json). The simpler version (Spot Market Chatbot.json) is more responsive to user's request. *The chatbot is already connected to the simpler version that is online on Azure.* If you want to also try the chatbot with the IAB categories support luis model, Follow the steps below:

1. in the intent.matches() function, There is a commented section as below:

```
/*for (var i = 0; i < iabCategoriesNumbers.length; i++) {
    itemEntityName = 'IAB' + iabCategoriesNumbers[i]+"::"+'IAB' +
iabCategoriesNumbers[i] + ' - item';
    var itemEntity = builder.EntityRecognizer.findEntity(args.entities,
itemEntityName);
    if (itemEntity) {
        break;
    }
}*/
```

Uncomment this section, and comment the line below:

```
var itemEntity = builder.EntityRecognizer.findEntity(args.entities, 'Item');
```

2. In the bot setup section, you can see LUIS recognizer section. change this line:

```
var luisModelURL = process.env.LUIS_MODEL_URL;
```

into this line:

```
var luisModelURL = process.env.LUIS_MODEL_WITH_CATEGORIES_URL;
```

After these two simple steps, redeploy the chatbot locally, and try some of the conversations again. You can see that There is a lower recognition support, as LUIS does not seem to be able to handle multiple entities (IAB categories) as easy as a generic entity such as "item".

ChatBot

1. Install node.js and NPM on the target server and note the server's public URL.
2. Install all required dependencies by running npm install on the directory root.
3. Register a new Bot with the Microsoft Bot Framework at <http://dev.botframework.com/>.
4. Specify the messaging endpoint as the server's URL, and register for an App ID and password pair from Microsoft via the same link.
5. Open the .env file in the project's root directory and specify the app ID and password registered above. Specify the address of the LUIS recognizer set up in the previous section. Also specify the URLs to the endpoints for the API.
6. Start the bot with the node app command.
7. If you want to test the bot locally, you can easily do that by downloading Microsoft bot emulator from <https://github.com/Microsoft/BotFramework-Emulator>, choosing the host the chatbot is running on and testing it.