GHC INDIA

Build Your Own Bot, Hands On

Document 1:MANUAL

Ravichandran, Charanya & Prabhu, Sharada

12/08/2016

Table of Contents

Pre-requisites

1. NodeJS installation

Visit NodeJS official site https://nodejs.org/en/ and install the latest version of nodejs based on your machine configuration.

2. Microsoft account

Create a Microsoft account by visiting https://www.microsoft.com/en-us/account/default.aspx

3. Download Bot Emulator for testing the bot locally. Follow instructions from the official site https://github.com/Microsoft/BotFramework-Emulator/blob/master/README.md

4. Code can be downloaded from: https://github.com/CharanyaR/firstbot

Preface

This manual is designed to provide instructions on building a Chat Bot named **Chef Bot** using Microsoft Bot Framework in NodeJS. It also describes the steps to test the developed Bot on the Skype platform or using Emulator or using web chat from Microsoft developer portal.

Step 1 – HelloWorld Chat Bot.

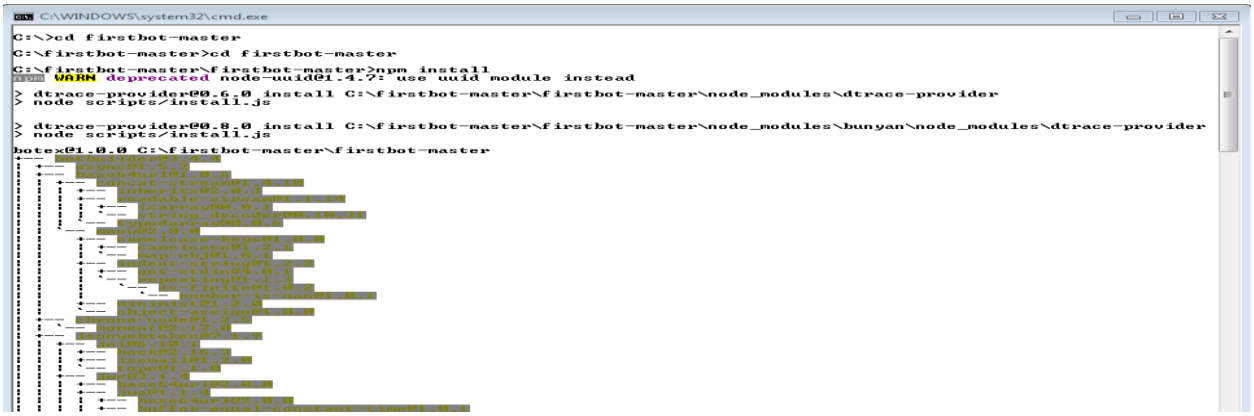
1. Go to command prompt.
2. Traverse to the directory where downloaded code exists:
   cd firstbot-master
3. Execute below command:
   npm install
4. Run **node HelloWorld.js**
   This will run the bot service in 8080 port.

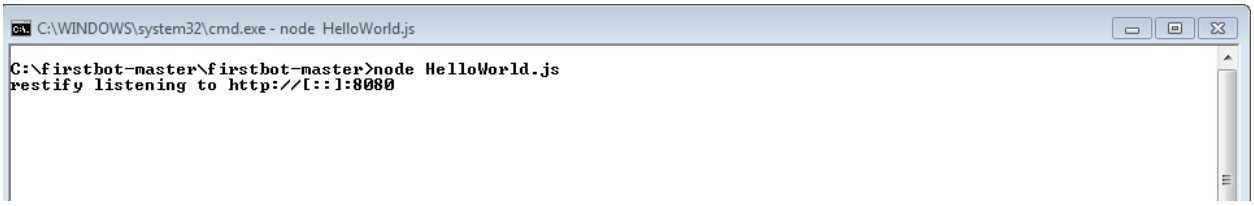
5. Open your Microsoft Bot Emulator. Test the bot locally by providing the Bot URL as: **http://localhost:8080/api/messages**

   When you enter any text to the bot it would respond "Hello World"

Code used:

https://github.com/CharanyaR/firstbot -> HelloWorld.js





Step 2 – Build a NLP Model using LUIS.

Use Case: Chef Bot.

When a user inputs an ingredient, the bot would respond with possible recipes that the user can try.

To build the Chef Bot we need some Artificial Intelligence added to the Chat bot. Microsoft's Language Understanding Intelligent Service (LUIS) offers a fast and effective way of adding language understanding to applications. LUIS model will be built to add natural language processing in our bot

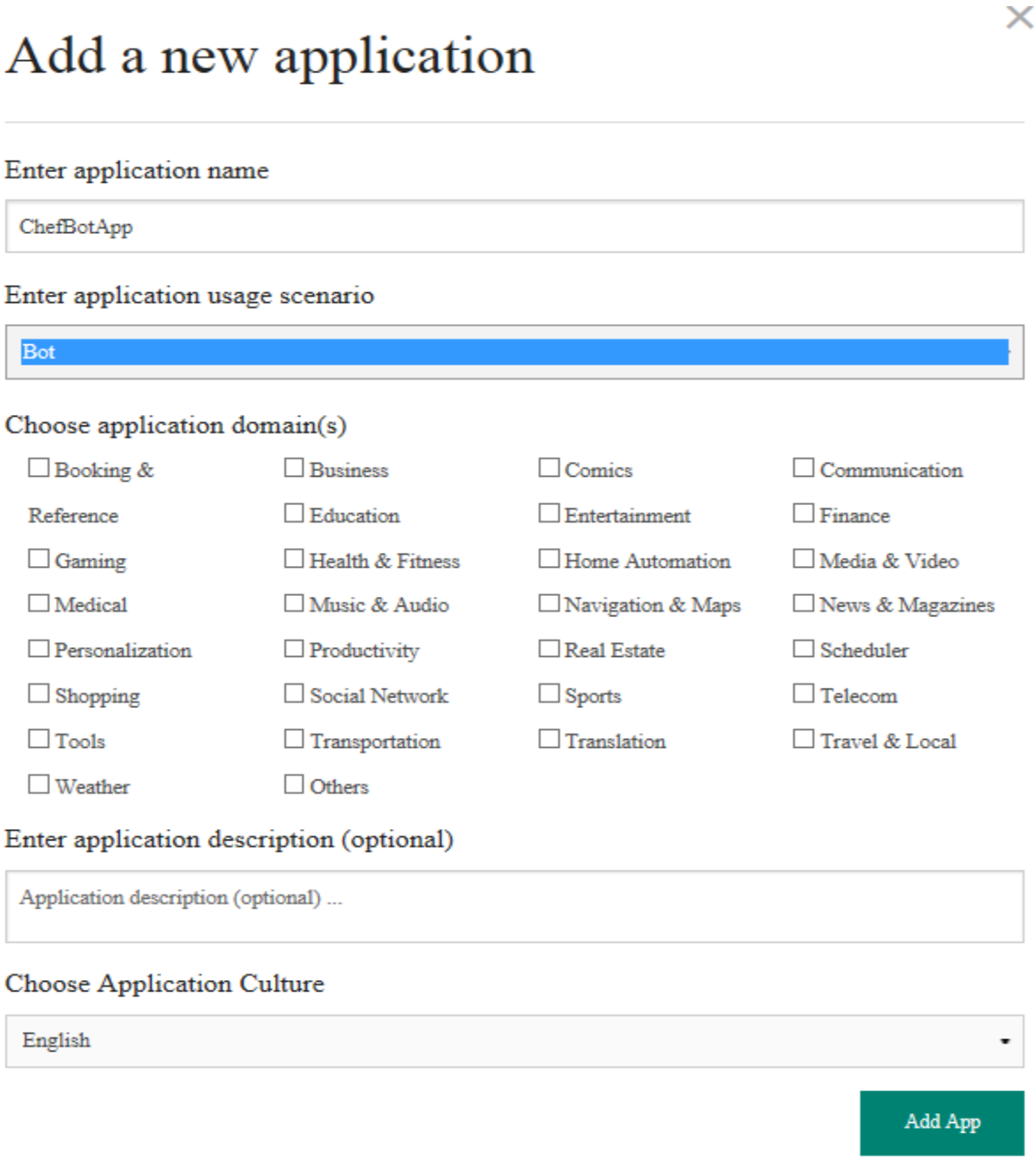
Steps:

1. Log into https://www.luis.ai/ using Microsoft account.

2. Click on New App icon after login.

3. In the popup enter a name for your application

Enter application name - ChefBotApp
Choose application usage scenario – Bot
Choose application domain - Entertainment
Choose Application Culture - English
Enter Add app.



Disclaimer: The views or opinions expressed in this presentation are solely those of the author and do not necessarily represent those of Fidelity Investments. This research does not reflect in any way procedures, processes or policies of operations within Fidelity.

4. Now the app is created. For our use case, Intent is "GetRecipe" and the Entity is "ingredients".

5. In the LUIS ChefBotApp created,
Add Intents and Entities:

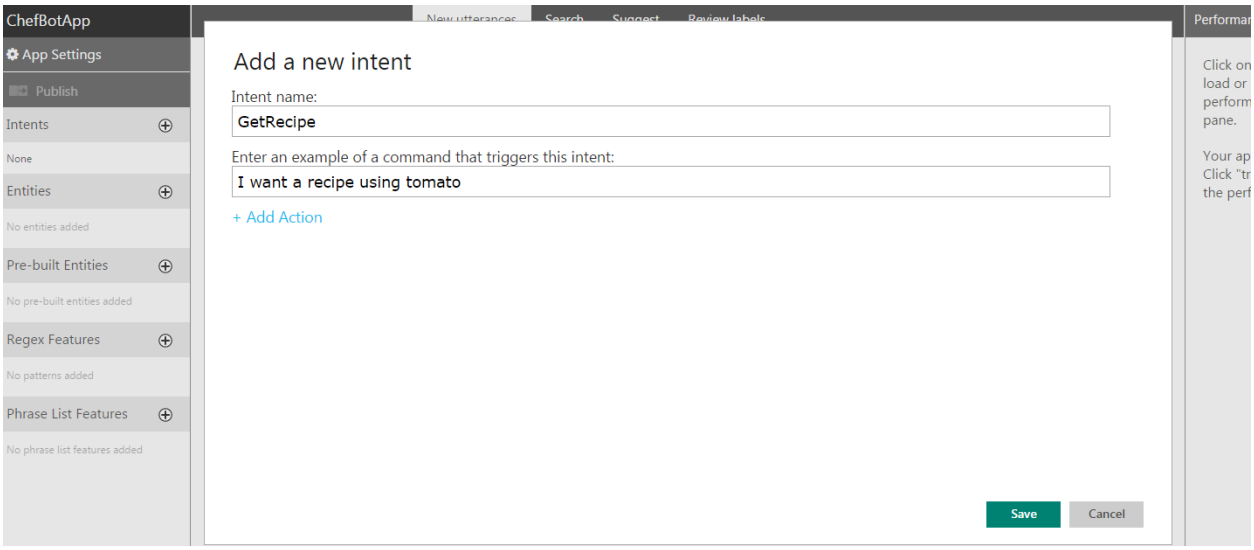
Intents

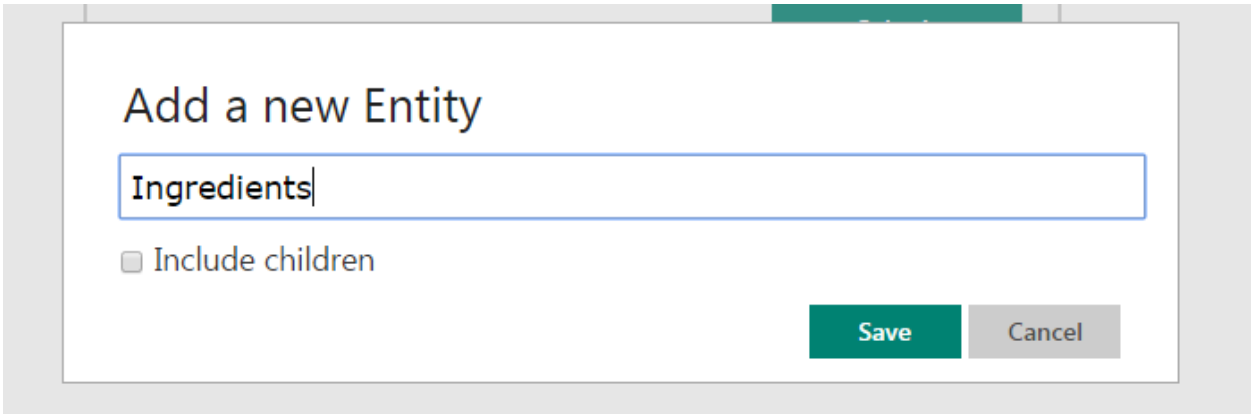
Create intent, name it GetRecipe.

Example usage - What recipe can I prepare with potato?

Entities

Create an entity, name it ingredients





Disclaimer: The views or opinions expressed in this presentation are solely those of the author and do not necessarily represent those of Fidelity Investments. This research does not reflect in any way procedures, processes or policies of operations within Fidelity.

6. Now our intent and entity ready. Let's add some utterances and start training our model.

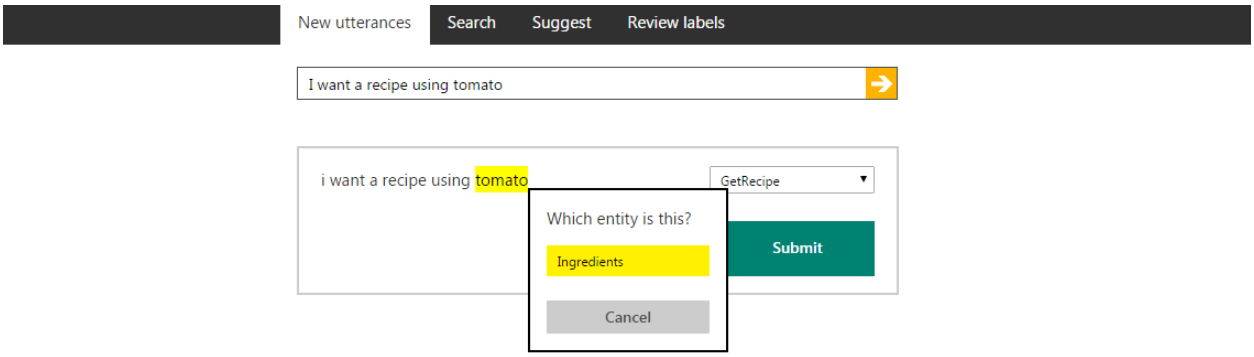
7. Example utterances try in the new utterances tab.

Can you suggest a recipe using tomato, garlic and onion?

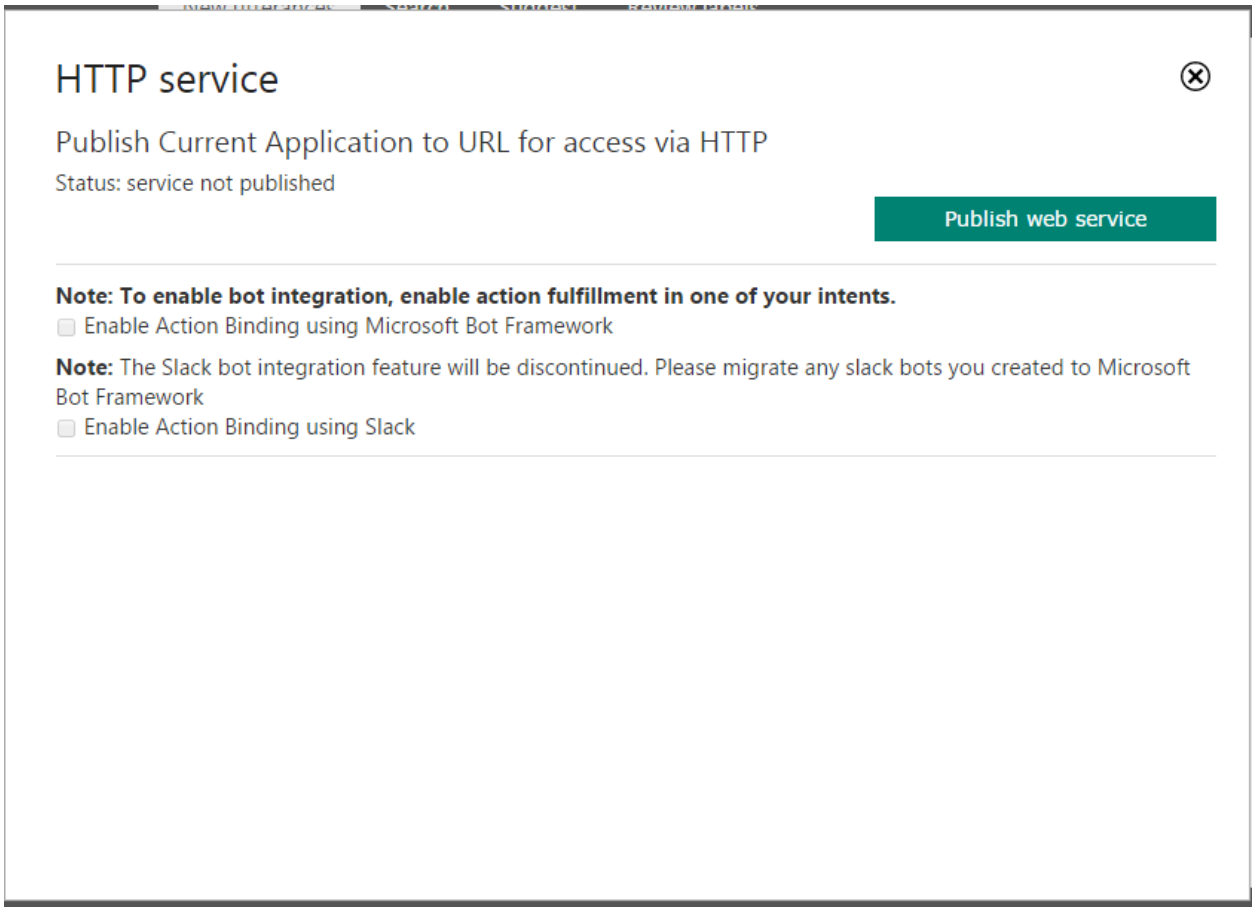
What can I prepare with potato?

Corn and avocado

Post adding these utterances and mapping them to the intent GetRecipe and entity ingredients. Click on train to train the model.



8. Now our LUIS model is ready. Let's publish it to consume as a service. Click on publish.



Note the LUIS model URL and the subscription key. Ex: https://api.projectoxford.ai/luis/v2.0/apps/7c8788eb-1e3f-46d9-9283-fc24f3f164c9?subscription-key=KEY_YOU_OBTAIN&verbose=true &q=

Step 3 – Create a client for Food2Fork.

To get recipe data we would be using the API from food2fork.

Search API which accepts ingredients to query for.

http://food2fork.com/api/search?key={API_KEY}&q=shredded%20chicken

API_KEY is the developer key we get on signup.

1. We will use the below file, which will have client code to call the food2fork API

https://github.com/CharanyaR/firstbot -> food2forkclient.js

Step 4 – Integrate all and test the ChefBot.

1. Now we have NLP model ready we will integrate it to our Bot.
   https://github.com/CharanyaR/firstbot -> index.js
2. Stop the node server. Give **Ctrl+C** in the command prompt where Node server is running.
3. Execute the command : **node index.js**
4. Now the bot service is started.
   Open your Microsoft Bot Emulator. Test the bot locally by providing the Bot URL as: **http://localhost:8080/api/messages**

Tunnel the locally running bot:

1. Run ngrok http 8080

This would generate a URL which will help you tunnel the locally running bot. This URL appended with end point need to be added while registering bot.

OR

2. Localtunnel lt --port 8080 [this would give a URL as well]

Register Bot in the Microsoft Dev Portal:

1. Log in to https://dev.botframework.com/

2. Click on Register a Bot tab.

Name: ChefBot

Bot handle: bot_1234

Description: Any description

Messaging endpoint: https://firstbot-charr.herokuapp.com/api/messages

Post this click on create Microsoft app id and password. Note down the password generated.

3. In our index.js add these appid and password.

4. Click on Test to test the connection with our Bot.

5. Using Web Chat you can start testing the bot.

Platform integration:

In the bot page in developer portal under our bot there is an option for add to Skype. By clicking that we can add our bot as a contact in our Skype account.

Reference:

Our GitHub repository:

https://github.com/CharanyaR/firstbot.git

Contact Mail-id:

Charanya, Ravichandran – ravichandran.charanya@gmail.com

Sharada, Prabhu – Sharada.prabhu@fmr.com