

Crispy The Bot

Sawarin Sodsangsook

5681091

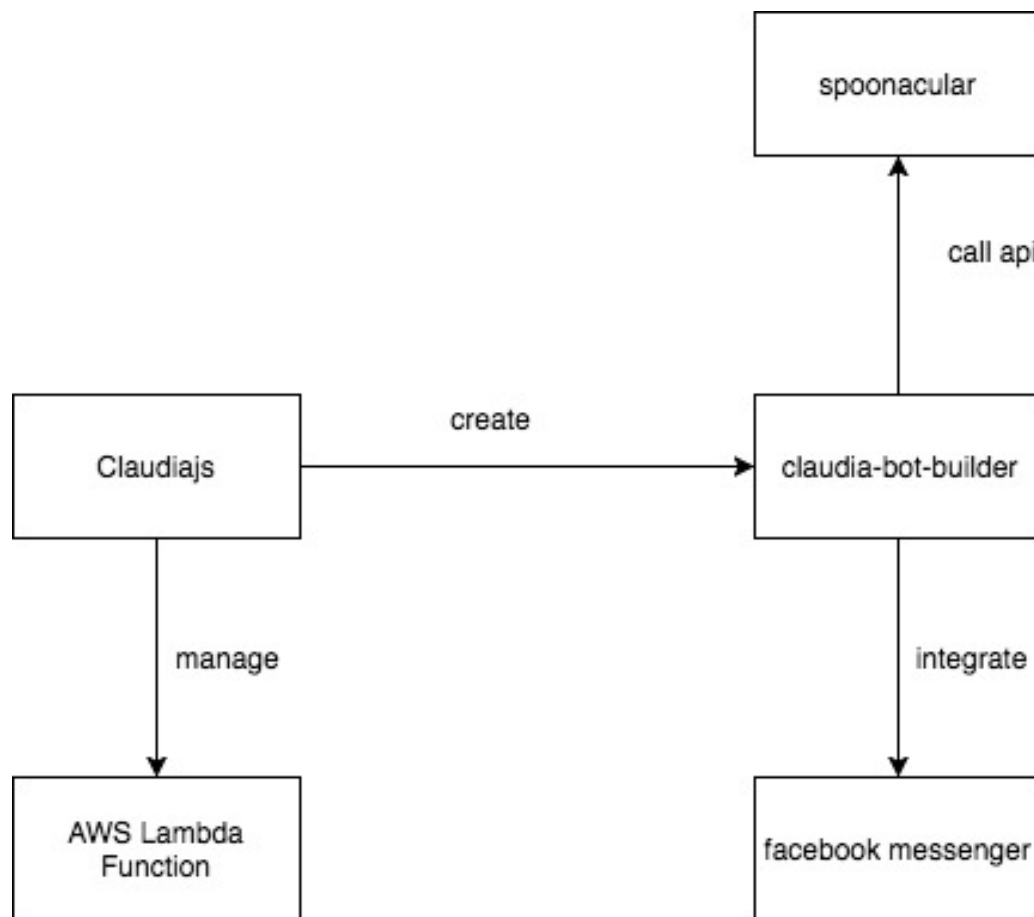
Final Project Report of Crispy The bot

This is the report of final project in subject ICCS486 Professional Practices on Network Technology I of Mahidol University International College, Third trimester 2016.

This project aims to support certain commands such as searching recipes and telling its details such as ingredient and instructions.

Software components:

- Claudiajs
to manage AWS Lambda Function
- spoonacular
for researching recipe
<https://market.mashape.com/spoonacular/recipe-food-nutrition#>
- claudia-bot-builder
to integrate with facebook messenger
- Bludebird
return a promise from api call
- Unirest
use to fetch a data from api



How my code works:

api.js : manage api call

3 functions

findEntities : Send the keyword to query relevant recipes

entityClaims : Send the id of a recipe to get detail such as

ingredients, instructions

findInText : Send a text to detect the keyword in text such as

dishes and ingredient

bot.js : Export actual bot as a module

do 5 actions:

- Find recipe => request.text.startwith"find"

call api.findEntities get res and make it in platform for user to choose which recipe they want. When they choose a recipe, module will be call again but request.text will be "GET {id f the recipe}"

- Claims the recipe => request.text.startwith"GET"

Call entityClaims with it id then make the options for user to choose whether see ingredients, instructions, or get a new recipe

See ingredient

module will be call again but request.text will be "ingre-{id f the recipe}"

See instructions

module will be call again but request.text will be "inst-{id f the recipe}"

get a new recipe

module will be call again but request.text will be '{keyword}' so that it won't go to do another action

- Get ingredient => request.text.startwith"ingre-"

Call entityClaim and display the ingredients

- Get instructions => request.text.startwith"inst-"

Call entityClaim and display the instructions

- Detect the keyword in text

If the request.text won't go to any of previous action it will first call api.findInText to see if any of keyword in request.text can be use EX if user input I like taco, salmon and nuts this will detect taco, salmon and

nuts then display the options of those keyword for user to select if the user selected, module will be call again but request.text will be find {that keyword}

But if no word is detected if will send the message to tell user to input any word.

package.json : store dependencies of the project

