SlackBot プログラムの仕様書

2016 年 04 月 21 日
乃村研究室 神澤 宏貴

1 概要

本資料は平成 28 年度 GN グループ B4 新人研修課題の SlackBot の仕様についてまとめたものである．本プログラムは以下の 2 つの機能をもつ．

(1) “「○○」と言って” という文字列を含むメッセージに対し，“○○” と発言する機能

(2) 文字列と翻訳規則を含むメッセージに対し，文字列を翻訳して発言する機能

なお，いずれの場合も，メッセージの先頭に “@K-Bot” が付いていることを前提としている．

2 対象とする使用者

本プログラムは以下の 3 つのアカウントを所有する利用者を対象としている．

(1) Slack アカウント

(2) Microsoft アカウント

3 機能

本プログラムが持つ 2 つの機能について述べる．

(機能 1) “「○○」と言って” という文字列を含むメッセージに対し，適切に発言する機能
この機能は，メッセージ内に “「○○」と言って” という文字列があった場合，“○○” と発言する機能である．また，“「「○○」と言って」と言って” のように入れ子構造になっている場合は，最長の文字列を発言する．上記の例の場合 “「○○」と言って” と発言する．

表 1: 言語対応表

言語	入力する文字
日本語	日
英語	英
フランス語	仏
スペイン語	西
ロシア語	露
アラビア語	アラビア

(機能 2) 文字列と翻訳規則を含むメッセージに対して，文字列を翻訳して発言する機能

この機能は，受信したメッセージに翻訳の要求があった場合，翻訳規則に沿って文字列を翻訳して発言する機能である．なお，翻訳には，Microsoft Translator API[1] を用いている．翻訳の要求の入力規則を以下に示す．

@K-Bot 翻訳 (1) → (2) “(3)”

(1) 翻訳前の言語

(2) 翻訳後の言語

(3) 翻訳してほしい文字列

ただし，(1)，(2) に指定可能な言語は表 1 の 6 つである．以下に翻訳の要求の例を示す．

@K-Bot 翻訳　日→英 “ありがとう”

この例では，Bot は “Thank you” と発言する．

なお，(機能 1)，(機能 2) の 2 つに対応するメッセージ以外のメッセージを受信した場合は，いずれも “Hi” と発言する．

4 動作環境

本プログラムの動作環境を表 2 に示す．また，表 2 に示す動作環境で本プログラムが正しく動作することを確認した．

表 2: 動作環境

項目	内容
OS	Linux Debian GNU/Linux(version 8.1)
CPU	Intel(R) Core(TM) i5-4590 CPU(3.30GHz)
メモリ	1.00GB
ブラウザ	Firefox 45.0.2
ソフトウェア	Ruby 2.1.5
	bundler 1.11.2
	git 2.1.4
	heroku-toolbelt 3.42.47

5 環境構築

5.1 概要

本プログラムを実行するために必要な環境構築の手順を以下に示す.

(1) Incoming WebHooks の設定

(2) Outgoing WebHooks の設定

(3) Heroku の設定

(4) Microsoft Translator API の設定

それぞれの手順について，5.2 節に記述する.

5.2 具体的な手順

5.2.1 Incoming Webhooks の設定

(1) 自身の Slack アカウントにログインする.

(2) Custom Integrations(https://xxxxx.slack.com/apps/manage/custom-integrations) へアクセスする．ただし xxxxx はチーム名である.

(3) 「Incoming WebHooks」をクリックする.

(4) 「Add Configuration」から，新たな Incoming WebHooks を追加する.

(5) Bot が発言するチャンネルを選択後,「Add Incoming WebHooks integration」をクリックし，Webhook URL を取得する.

5.2.2 Outgoing Webhooks の設定

(1) Custom Integrations にアクセスする.

(2) 「Outgoing WebHooks」をクリックする.

(3) 「Add configuration」から，新たな Outgoing WebHook を追加する.

(4) 「Add Outgoing WebHooks integration」をクリックする.

(5) Outgoing WebHooks に関して以下を設定する.

(A) Channel: 発言を監視する channel
(B) Trigger Word(s): WebHook が動作する契機となる単語 (@K-Bot)
(C) URL(s): WebHook が動作した際に POST を行う URL

5.2.3 Heroku の設定

(1) Heroku(https://dashboard.heroku.com/) にアクセスする.

(2) 「Sign up」から新しいアカウントを登録する.

(3) Heroku から送信されたメールのに記載されている URL をクリックし，パスワードを設定する.

(4) 登録したアカウントでログインし,「Getting Started with Heroku」の使用する言語で Ruby を選択する.

(5) 「I'm ready to start」をクリックし,「Download Heroku Toolbelt for...」から Toolbelt をダウンロードする.

(6) ターミナルで以下のコマンドを実行し，Toolbelt がインストールされたことを確認する.

```
$ heroku version
```

(7) 以下のコマンドを実行し，Heroku にログインする.

```
$ heroku login
```

(8) 本プログラムがあるディレクトリに移動する．

(9) Heroku 上にアプリケーションを生成するための以下のコマンドを実行する．

$ heroku create xxxxx

ただし，xxxxx はアプリケーション名である．アプリケーション名は小文字と数字，およびハイフンのみ使用できる．

(10) 以下のコマンドを実行し，アプリケーションが生成されていることを確認する．

$ git remote -v

なお，アプリケーションが生成されることは，以下の文字列が表示されることで確認できる

heroku https://git.heroku.com/xxxxx.git (fetch)
heroku https://git.heroku.com/xxxxx.git (push)

(11) 以下のコマンドを実行し，5.2.1 の (6) で取得した WebHook URL を，Heroku の環境変数に設定する．

$ heroku config:set INCOMING_WEBHOOK = “https://xxxxxxxx”

5.2.4 Microsoft Translator API の設定

(1) Microsoft Azure Marketplace(https://datamarket.azure.com/home) にアクセスする．

(2) 自身の Microsoft アカウントにサインインする．

(3) 「データ」をクリックする．

(4) 「Microsogt Translator - Text Translation」をクリックする．

(5) 「サインアップ」をクリックする．

(6) 以下の項目を設定する

(A) クライアント ID
(B) 名前
(C) 顧客の秘密

(D) リダイレクト URI
例：https://localhost/

(7) 設定したクライアント ID，顧客の秘密をそれぞれ MySlackBot.rb の MySlackBot クラス以下の CLIENT_ID，CLIENT_SECRET の欄に以下のように記述する．

CLEINT_ID = “(クライアント ID)”
CLIENT_SECRET = “(顧客の秘密)”

5.2.5 Gem のインストール

(1) Gem をインストールするために，以下のコマンドを実行する．

$ bundle install –path vendor/bundle

6 使用方法

本プログラムは，Heroku 上で実行することを想定している．このため，Heroku へデプロイするとプログラムは実行状態となる．Heroku へデプロイする手順を以下にに示す．

(1) 以下のコマンドを実行し，変更をコミットする．

$ git add MySlackBot.rb
$ git commit

(2) 以下のコマンドを実行し，heroku にプッシュする．

$ git push heroku master

7 エラー処理と保証しない動作

7.1 エラー処理

本プログラムのエラー処理を以下に示す．

(1) (機能 2) において以下の 2 つのエラー処理を実装した．

(A)　表 1 に示していない言語が指定された場合，Bot は以下の発言をする．
対応している言語は以下の言語です．
日本語：日
英語：英
フランス語：仏
スペイン語：西
ロシア語：露
アラビア語：アラビア

(B) メッセージのフォーマットが翻訳の要求に沿っていない場合，Bot は以下の発言をする．
翻訳のフォーマットは以下のとおりです．
翻訳 翻訳前の言語→翻訳後の言語 “文字列”

7.2　保証しない動作

本プログラムの保証しない動作を以下に示す．

(1) 表 2 に示す動作環境以外でプログラムを実行

(2) Slack の Outgoing WebHooks 以外から POST リクエストを受け取ったときの処理

参考文献

[1] Microsoft: Microsoft Translator - Text Translation, Microsoft (online), available from ⟨https://datamarket.azure.com/dataset/bing/microsofttranslator⟩ (accessed 2016-04-21).