SlackBot プログラムの仕様書

2018/4/25

藤原 裕貴

1 概要

本資料は，2018 年度 B4 新人研修課題にて作成した SlackBot プログラムの仕様についてまとめたものである．本プログラムで使用する Slack[1] とは Web 上で利用できるチームコミュニケーションツールである．SlackBot とはある契機により自動で Slack に発言するプログラムのことである．本プログラムは，以下の 2 つの機能を持つ．

(1) 指定された文字列を発言
(2) 指定された移動手段と 2 つの地点から 2 つの地点間の距離，移動にかかる時間，および経路を発言

なお，本資料においての発言とは Slack の特定のチャンネル上で発言することを指す．また，本資料においての発言内容は “”で囲って表す．

2 対象とする利用者

本プログラムは，以下の 2 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) Google アカウント

Google アカウントは本プログラムで使用する Google Maps API のキーを取得するのに必要である．

3 機能

本プログラムは Slack での “@FBot”から始まるユーザの発言を受信し，これに対して SlackBot が発言する． SlackBot から発言される内容は “@FBot”に続く文字列により決定される．以下に本プログラムが持つ 2 つの機能について述べる．

(機能 1) 指定された文字列を発言

この機能はユーザの “@FBot 「(任意の文字列)」と言って”という発言に対して，「」内の文字列をユーザに発言する．たとえば，“@FBot 「こんにちは」と言って”という発言に対して SlackBot は “こんにちは”と発言する．「」が重複する際，SlackBot は「」内が最も長くなるように発言する．たとえば，“@FBot 「「こんにちは」と言って」と言って”という発言に対して SlackBot は “「こんにちは」と言って”と発言する．

(機能 2) 指定された移動手段と 2 つの地点から 2 つの地点間の距離，移動にかかる時間，および経路を発言

この機能はユーザの “@FBot (移動手段) での (出発地点) から (到着地点) までの道” という発言に対して，SlackBot は以下の情報を発言する．

(1) 出発地点から到着地点までの距離
(2) 出発地点から到着地点までの移動にかかる時間
(3) 出発地点から到着地点までの経路を示した Google Map へのリンク

なお，移動手段は徒歩，自動車，および電車から指定する．出発地点と到着地点は，場所の名称か住所で指定する．たとえば，ユーザの “@FBot 徒歩での岡山大学から岡山駅までの道” という発言に対して，SlackBot は，以下のように情報を発言する．

距離は `2.9 km`
かかる時間は `37` **分**
詳しい道はこちら

SlackBot が発言した “こちら” をクリックすることで詳しい経路を示した Google Map を見ることができる．

これらの情報は，住所を緯度と経度に変換する Google Maps Geocoding API[2] と移動手段，緯度，および経度から経路を作成する Google Maps Directions API[3] を使用して取得している．この 2 つの API は Google Maps API キーにより使用できる．

上記の (機能 1) と (機能 2) のどちらにもあてはまるユーザの発言を受信したときは，(機能 1) が優先される．

上記の (機能 1) と (機能 2) のどちらにもあてはまらないユーザの発言を受信したときは，SlackBot は以下を発言する．

「〇〇」と言って `or` **（移動手段）での（出発地点）から（到着地点）までの道**

4 動作環境

本プログラムの動作環境を表 1 に示す．なお，表 1 での動作環境では動作を確認済みである．

5 環境構築

5.1 概要

本プログラムの動作のために必要な環境構築の項目を以下に示す．

(1) Slack の Incoming WebHooks の設定

表 1　動作環境

項目	内容
OS	Linux Debian GNU/Linux 8
CPU	Intel(R) Core(TM) i5-4590 CPU (3.30GHz)
メモリ	1.0GB
Ruby	ruby 2.5.1
Ruby Gem	bundler 1.16.1
	mustermann 1.0.2
	rack 2.0.4
	rack-protection 2.0.1
	tilt 2.0.8
	sinatra 2.0.1

(2) Slack の Outgoing WebHooks の設定
(3) Heroku の設定
(4) Google Maps API の API キー取得

次節で各項目の手順を述べる．

5.2 具体的な手順

5.2.1 Slack の Incoming WebHooks の設定

Slack が提供する Incoming WebHooks の設定を行う．Incoming WebHooks とは，指定したチャンネルの URL に発言を POST する機能である．以下に設定手順を示す．

(1) 以下の URL にアクセスする．
`https://<team_name>.slack.com/apps/manage/custom-integrations`
ただし，`<team_name>`は Slack で所属するチーム名に置き換える．
(2)「Incoming WebHooks」をクリックする．
(3)「Add Configuration」をクリックする．
(4) 発言先のチャンネルを選択する．
(5)「Add Incoming WebHooks integration」をクリックすることで WebHook URL を取得する．

5.2.2 Slack の Outgoing WebHooks の設定

Slack が提供する Outgoing WebHooks の設定を行う．Outgoing WebHooks とは，指定したチャンネルでの発言に指定した文字列が含まれているとき，指定した URL にその情報を POST する機能である．以下に設定手順を示す．

表 2　Outgoing WebHooks の設定項目

項目	内容
Channel	発言を取得したいチャンネルを設定
Trigger Word(s)	Outgoing WebHooks が動作する契機となる単語を設定
URL(s)	Outgoing WebHooks が動作した際に POST を行う URL

(1) 以下の URL にアクセスする．
`https://<team_name>.slack.com/apps/manage/custom-integrations`
ただし，`<team_name>`は Slack で所属するチーム名に置き換える．

(2)「Outgoing WebHooks」をクリックする．
(3)「Add Outgoing WebHooks integration」をクリックする．
(4) Outgoing WebHooks の各項目を設定する．設定する項目は表 2 に示す．
なお，表 2 の URL(s) には `https://<my_app_name>.herokuapp.com/slack` を設定する．ただし，`<my_app_name>`は 5.2.3 項 (5) にて指定するアプリケーション名である．

5.2.3　Heroku の設定

Heroku とは，PaaS(Platform as a Service) と呼ばれるサービスであり，Web アプリケーションを実行するためのプラットフォームである．以下に設定手順を示す．

(1) 以下の URL より Heroku にアクセスし，「Sing up」から新しいアカウントを登録する．
`https://www.heroku.com/`
(2) 登録したアカウントでログインし，「Getting Started with Heroku」の使用する言語として「Ruby」を選択する．
(3)「I'm ready to start」をクリックし，「Download Heroku CLI for...」から CLI をダウンロードする．
(4) ターミナルで以下のコマンドを実行し Heroku にログインする．
`$ heroku login`
(5) 以下のコマンドを実行し Heroku 上にアプリケーションを生成する．
`$ heroku create <my_app_name>`
ここで，`<my_app_name>`は作成するアプリケーション名である．アプリケーション名はアルファベット小文字，数字，およびハイフンのみ使用できる．
(6) 以下のコマンドを実行し，Heroku の環境変数に WebHook URL を設定する．
`$ heroku config:set INCOMING_WEBHOOK_URL="<webhook_url>"`
ただし，`<webhook_url>`は 5.2.1 項 (5) で取得した WebHook URL に置き換える．

5.2.4 Google Maps API の API キー取得

(1) 以下の URL にアクセスし,「キーの取得」をクリックする.
`https://developers.google.com/maps/web/`

(2)「Select or create project」をクリックする.

(3)「Create a new project」をクリックし，プロジェクト名を付ける.

(4)「NEXT」をクリックすると，Google Maps API キーが作成される.

作成されたキーを用いることで，Google Maps Geocoding API と Google Maps Directions API が使用できる.

6 使用方法

本プログラムの使用方法について述べる．本プログラムは Heroku にデプロイすることにより実行できる．デプロイには，以下のコマンドを実行する.

```
$ 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

本プログラムで行ったエラー処理を以下に示す.

(1) (機能 2) について，交通手段に徒歩，自動車，および電車以外を指定した場合，以下のように発言する.

交通手段は徒歩か自動車か電車を選んでください

(2) (機能 2) について，Google Maps Geocoding API で指定した地点が発見されず，緯度と経度に変換できなかった場合と Google Maps Directions API で経路を発見できなかった場合，以下のように発言する.

測定できませんでした

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) Slack の Outgoing WebHooks 以外からの POST リクエストをブロックする動作.

(2) (機能 2) において，指定する地点の名前が正しくない場合の動作.

参考文献

[1] Slack: Where work happens, Slack (online), available from ⟨https://slack.com/⟩ (accessed 2018-04-25).

[2] Google Inc.: Google Maps Geocoding API, Google (online), available from ⟨https://developers.google.com/maps/documentation/geocoding/start⟩ (accessed 2018-04-25).

[3] Google Inc.: Google Maps Directions API, Google (online), available from ⟨https://developers.google.com/maps/documentation/directions/⟩ (accessed 2018-04-25).