SlackBot プログラム 仕様書

2018/4/24

高橋 桃花

1 概要

本資料は，平成 30 年度 B4 新人研修課題の 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，チャットツールである．SlackBot とは，Slack におけるやり取りを自動化するプログラムである．本プログラムは以下の 2 つの機能をもつ．

(1) 入力された文字列を返信する機能
(2) 入力された飲食店の情報を返信する機能

また，本資料において引用符 “” に囲まれた文字列は，Slack における発言を示す．

2 対象とする利用者

本プログラムは以下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 API キーの取得に必要である．

3 機能

本プログラムは，Slack での “@TakaBot” という文字列から始まる発言を受信し，その発言に対して返信する．返信の内容は “@TakaBot” 以降の文字列により決定される．以下に本プログラムがもつ 2 つの機能について述べる．

(機能 1) 入力された文字列を返信する機能

ユーザが “@TakaBot 「(指定された文字列)」と言って” と発言した場合，本プログラムは鉤括弧内の文字列を返信する．例えば，ユーザが “@TakaBot 「こんにちは」と言って” と発言した場合，本プログラムは “こんにちは” と返信する．

(機能 2) 入力された飲食店の情報を返信する機能

ユーザが “「(飲食店の名前)」の情報” と発言した場合，本プログラムは鉤括弧内で指定された飲食店の詳細情報を返信する．詳細情報は以下の通りである．

(1) 飲食店の名前
(2) 開店ステータス
(3) 価格帯
(4) 評価
(5) URL
(6) 最新のレビュー
(7) 投稿写真

以上の情報は Google Places API[2] を利用して取得している．

上記の (機能 1)，(機能 2) のどちらにも当てはまらない発言を受信した場合，本プログラムは以下のように返信する．

@ユーザ名 `Hi!`
使い方 1：「○○」と言って，使い方 2：「飲食店の名前」の情報

4 動作環境

本プログラムの動作環境を表 1 に示す．また，表 1 の環境において本プログラムが正常に動作することを確認した．

表 1 動作環境

項目	内容
OS	Debian GNU/Linux 8 (jessie) 64-bit
CPU	Intel Core i5-4670 CPU (3.40GHz)
メモリ	1.0GB
ブラウザ	FireFox 59.0.2
ソフトウェア	Ruby 2.5.1
	bundler 1.16.1
	heroku CLI 6.16.11-b6217f5
	Git 2.1.4

また，本プログラムに必要な Gem を表 2 に示す．Gem とは，Ruby で使用することのできるライブラリである．

表 2　本プログラムに必要な Gem

Gem	バージョン
mustermann	1.0.2
rack	2.0.4
rack-protection	2.0.1
sinatra	2.0.1
tilt	2.0.8

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) Heroku の設定
(2) Slack の Incoming WebHooks の設定
(3) Slack の Outgoing WebHooks の設定
(4) Google Places API の API キー取得
(5) Gem のインストール

次節で各項目における具体的な手順について述べる.

5.2　手順

5.2.1　Heroku の設定

(1) 以下の URL より Heroku にアクセスし,「Sign 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（Command Line Interface）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 CLI がインストールされたことを確認する.

```
$ heroku version
heroku-cli/6.16.11-b6217f5 (linux-x64) node-v9.11.1
```

(5) 以下のコマンドを実行し，Heroku にログインする.

```
$ heroku login
```

(6) 本プログラムのディレクトリに移動して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 <myapp_name>
```

ただし，<myapp_name>は任意のアプリケーション名を示す．アプリケーション名には小文字，数字，およびハイフンのみ使用できる．

(7) 以下のコマンドを実行し，生成したアプリケーションがリモートリポジトリに登録されていることを確認する．

```
$ git remote -v
heroku https://git.heroku.com/<myapp_name>.git (fetch)
heroku https://git.heroku.com/<myapp_name>.git (push)
```

5.2.2 Slack の Incoming WebHooks の設定

(1) 以下の URL にアクセスする．
https://XXXXX.slack.com/apps/manage/custom-integrations
ただし，XXXXX は自分のチーム名を示す．

(2)「Incoming WebHooks」をクリックする．

(3)「Add Configuration」をクリックする．

(4)「Choose a channel...」から発言を投稿したいチャンネルを選択し，「Add Incoming WebHooks integration」をクリックする．

(5) WebHook URL を取得する．以下のコマンドにより取得した URL を Heroku の環境変数に設定する．

```
$ heroku config:set INCOMING_WEBHOOK_URL="https://XXXXX"
```

ただし，XXXXX は取得した自分の WebHook URL を示す．

5.2.3 Slack の Outgoing WebHooks の設定

(1) 以下の URL にアクセスする．
https://XXXXX.slack.com/apps/manage/custom-integrations
ただし，XXXXX は自分のチーム名を示す．

(2)「Outgoing WebHooks」をクリックする．

(3)「Add Configuration」をクリックする．

(4)「Add Outgoing WebHooks integration」をクリックし，以下の項目を設定する．

（A）Channel にて，発言を監視するチャンネルを選択する．
（B）Trigger Word(s) に，WebHooks が動作する契機となる単語を設定する．
（C）URL(s) に，WebHooks が動作した際に POST する URL を設定する．本プログラムは Heroku 上で動作させるため以下の URL を設定する．
https://XXXXX.herokuapp.com/slack
ただし，XXXXX は Heroku に登録したアプリケーション名を示す．

5.2.4 Google Places 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/places/web-service
(2)「Create a new project」を選択し，プロジェクト名を決定する．
(3)「Next」をクリックすると API キーが生成される．
(4) 以下のコマンドを実行し，Heroku の環境変数に取得した API キーを設定する．

```
$ heroku config:set GOOGLE_PLACES_APIKEY="API key"
```

5.2.5 Gem のインストール

本プログラムで使用する Gem を bundler を用いてインストールする．bundler とは，Gem の依存関係やバージョンを管理するためのものである．

(1) 以下のコマンドを実行し，Gem をインストールする．

```
$ bundle install --path vendor/bundle
```

6 使用方法

本プログラムの仕様方法について述べる．本プログラムは Heroku 上で動作するため，Heroku へデプロイすることで実行できる．以下のコマンドを用いて Heroku にデプロイする．

```
$ git push heroku master
```

7　エラー処理と保証しない動作

本プログラムにおけるエラー処理と保証しない動作について述べる．

7.1　エラー処理

(1) (機能 2) について，本プログラムが指定された飲食店の情報を Google Places API から取得できなかった場合，以下のようにユーザに返信する．

結果が取得できませんでした

7.2　保証しない動作

本プログラムが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

参考文献

[1] Slack Technologies, Inc.: Slack, Slack (online), available from ⟨https://slack.com/⟩ (accessed 2018-4-20).

[2] Google, Inc.: Google Places API, Google (online), available from ⟨https://developers.google.com/places/⟩ (accessed 2018-4-20).