SlackBot プログラム 仕様書

2018/5/7

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 のチャットにおいて発言したり，ユーザが特定の文字列を入力すると自動で返信したり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

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．また，本資料における返信とは，ユーザの発言に対し，本プログラムが応答することを示す．

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，こ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．たとえば，ユーザが “@TakaBot 「こんにちは」と言って” と発言した場合，本プログラムは図 1 のように返信する．

(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 “@TakaBot 「(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入力された飲食店の詳細情報を返信する．ただし，入力された飲食店の場所を特定するため，入力

図 1 “@TakaBot 「こんにちは」と言って” に対する返信

する文字列には，都道府県名，市町村名，駅名，および店舗名などを含む必要がある．飲食店の詳細情報は Google Places API[2] を利用して取得している．Google Places API により，Google マップ [3]，Google プレイス [4]，および Google+[5] のデータベースにアクセスし，データを取得できる．本機能により返信される内容の詳細は以下のとおりである．

(1) 飲食店の名前
入力された飲食店名から，最も適切な飲食店の情報を Google Places API により取得する．これにより，取得した飲食店の名前を表示する．

(2) 開店ステータス
ユーザが本プログラムに対する発言を入力した時点で，(1) の飲食店が営業中であるか休業中であるかを表示する．

(3) 価格帯
(1) の飲食店の価格帯を 0 から 4 の数字で表示する．詳細は以下のとおりである．
（A） 0: 無料
（B） 1: 安い
（C） 2: 普通
（D） 3: やや高い
（E） 4: 非常に高い

(4) 評価
Google プレイスのレビューの集計に基づき，(1) の飲食店の評価を表示する．評価は 1.0 から 5.0 の数字で表される．

(5) URL
(1) の飲食店の公式 Web サイトの URL を表示する．

(6) レビュー
Google Places API により取得した，(1) の飲食店に関するレビューを 1 件表示する．

(7) 写真
Google Places API により取得した，(1) の飲食店に関する写真を 1 枚表示する．

たとえば，ユーザが “@TakaBot 「スターバックス　岡山」の情報” と発言した場合，本プログラムは図 2 のように返信する．

図 2　“@TakaBot 「スターバックス　岡山」の情報” に対する返信

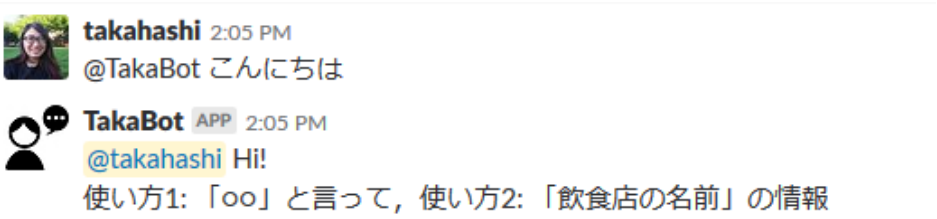

図 3　(機能 1)，(機能 2) のどちらにもあてはまらない場合

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，本プログラムは図 3 のように返信する．

4　動作環境

本プログラムは Heroku[6] 上で動作させる．Heroku とは，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．なお，本プログラムでは Heroku のフリープランを利用している．以下，表 1 に Heroku の環境を示す．

表 1　動作環境（Heroku）

項目	内容
OS	Ubuntu 16.04.4 LTS
CPU	Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz
メモリ	512MB
Ruby	2.5.1p57
Ruby Gem	bundler 1.16.1
	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

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 <app_name>
```

ただし，<app_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英語のアルファベットの小文字，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/<app_name>.git (fetch)
heroku https://git.heroku.com/<app_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://<team_name>.slack.com/apps/manage/custom-integrations
ただし，<team_name> 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://<webhook_url>"
```

ただし，<webhook_url>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://<team_name>.slack.com/apps/manage/custom-integrations
ただし，<team_name> 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://<app_name>.herokuapp.com/slack
ただし，<app_name> 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"
```

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 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，入力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，図 4 のようにユーザに返信する．

図 4　Google Places API から情報を取得できなかった場合

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).

[3] Google, Inc.: Google Maps, Google (online), available from ⟨https://www.google.com/maps/about/⟩ (accessed 2018-4-24).

[4] Google, Inc.: Google My Business, Google (online), available from ⟨https://www.google.com/business/⟩ (accessed 2018-4-24).

[5] Google, Inc.: Google+, Google (online), available from ⟨https://plus.google.com/about⟩ (accessed 2018-4-24).

[6] Heroku: Heroku, Salesforce.com, Inc. (online), available from ⟨https://www.heroku.com/about⟩ (accessed 2018-4-24).