SlackBot プログラムの仕様書

2016 年 4 月 21 日
乃村研究室 石川 大夢

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 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発言に反応して発言する機能

(2) 都道府県名と住所または駅名を含む発言に反応して bar を検索し，得られた検索結果を発言する機能

なお，本資料において発言とは，チャットツールである Slack の特定のチャンネル上で発言すること，または発言そのものを指す．また，本資料において，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

3　機能

本プログラムは，“@MoyaiBot ” という文字列から始まる発言を Bot へのリクエストメッセージとし，これに返信する．また，“@MoyaiBot ” 以降の内容によって Bot の発言の内容が決定される．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) “「○○○」と言って” という文字列を含む発言に反応して返信する機能
“@MoyaiBot ” という文字列の直後が “「○○○」と言って” という文字列であった場合，“○○○” と返信する．

(機能 2) 都道府県名と住所または駅名を含む発言に反応して bar を検索し，得られた検索結果を発言する機能
“都道府県名 (住所 or 駅名周辺) の bar” の形式に一致する文字列から始まる場

表 1: 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 バージョン 45.0.2
ソフトウェア	Ruby バージョン 2.1.5
	sinatra バージョン 1.4.7
	bundler バージョン 1.11.2
	Git バージョン 2.1.4

合，指定した都道府県の住所または駅付近の bar を検索し，得られた店の情報を返信する．発言に含まれる店の情報は，店名，営業時間，住所，および店内の画像である．Bot は最大で 10 件の bar の情報を発言する．また，リクエストのメッセージには都道府県の指定は必須であり，住所，駅名はどちらか一方の指定が必須である．以下に bar 検索リクエストの例を示す．

(例 1 住所指定の場合) @MoyaiBot 岡山県 岡山市北区の bar

(例 2 駅名指定の場合) @MoyaiBot 東京都 新宿駅周辺の bar

なお，上記の (機能 1)，(機能 2) のどちらにも当てはまらないリクエストメッセージを受信した場合は “Hi!” という文字列を発言する．

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 に示す動作環境での動作は確認済みである．

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) Slack アカウントの Incoming WebHooks の設定

(2) Slack アカウントの Outgoing WebHooks の設定

(3) SUNTORY BAR-NAVI WebAPI の API キー取得 [1]

(4) Heroku のアカウント作成と設定

(5) Gem のインストール

次節で具体的な手順を述べる.

5.2 具体的な手順

5.2.1 Slack アカウントの Incoming WebHooks の設定

Slack が提供している Incoming WebHooks の設定を行う．Incoming WebHooks とは指定したチャンネルへの URL に発言を POST する機能である.

(1) ブラウザを用いて，本プログラム使用者の Slack アカウントにログインする.

(2) ページ左上のチーム名をクリックし,「Apps & integrations」をクリックする.

(3) ページ右上の「Build your own」をクリックし,「Make a Custom Integration」をクリックする.

(4) 「Incoming WebHooks」をクリックする.

(5) 発言するチャンネルを選択し,「Add Incoming WebHooks integration」をクリックする.

(6) Webhook URL を取得する.

5.2.2 Slack アカウントの Outgoing WebHooks の設定

Slack が提供している 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 にその発言の情報を POST する機能である.

(1) 5.2.1 項の (3) まで同様の手順を踏む.

(2) 「Outgoing WebHooks」をクリックし,「Add Outgoing WebHooks integration」をクリックする.

表 2: Outgoing WebHooks の設定項目

項目	内容
Channel	発言を取得したい channel を指定
Trigger Word(s)	@MoyaiBot
URL(s)	https://<myapp_name>.herokuapp.com/slack myapp_name は 5.2.4 項 (9) にて指定するアプリケーション名

(3) Outgoing WebHooks のページの各設定項目を設定する．設定する項目は表 2 に示す.

(4) 「Save Settings」をクリックする.

5.2.3 SUNTORY BAR-NAVI WebAPI の API キー取得

SUNTORY が提供している bar 検索 WebAPI である SUNTORY BAR-NAVI WebAPI の API キーの取得を行う.

(1) 以下の URL から SUNTORY BAR-NAVI WebAPI のページにアクセスする.
`http://webapi.suntory.co.jp/barnavidocs/`

(2) ページ右側にある「メニュー」の「API キー取得」をクリックする.

(3) 登録するメールアドレスを入力し,「利用規約に同意して登録」をクリックする.

(4) 登録したメールアドレスに届いたメールにて API キーを取得.

5.2.4 Heroku のアカウント作成と設定

Web アプリケーションを開発するためのプラットフォームである Heroku のアカウント作成と使用するための設定を行う.

(1) 以下の URL から Heroku にアクセスする.
`https://www.heroku.com/`

(2) ページ右上の「Sign up」からアカウント作成画面へアクセスする.

(3) 必要項目を記入し,「Create Free Account」をクリックしアカウントを作成する.

(4) アカウント登録に使用したメールアドレスにHerokuからメールが送られてくるため，メールに記載されているURLにアクセスし．パスワードを設定する．

(5) 登録したアカウントでHerokuにログインし，「Getting Started with Heroku」からRubyを選択する．

(6) ページ下部の「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．

(7) 本プログラムが存在するディレクトリに移動する．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
```

(8) 以下のコマンドを実行し，Herokuにログインする．

```
$ heroku login
```

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

```
heroku create <myapp_name>
```

なお，myapp_nameは作成するアプリケーションの名前である．小文字，数字，およびハイフンのみ使用できる．

(10) 以下のコマンドで生成したアプリケーションがremote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)
```

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

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

(12) 以下のコマンドを実行し，Herokuの環境変数に5.2.3項(4)で取得したAPIキーを設定する．

```
$ heroku config:set BAR_API_KEY="*****"
```

5.2.5 Gem のインストール

本プログラム内で使用する Gem を bundler を用いてインストールする．Gem とは，Ruby で使用することができるライブラリである．

(1) 以下のコマンドを実行し，Gem を vendor/bundle 以下にインストールする．

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

6 使用方法

本プログラムを実行するための手順を以下に示す．本プログラムは Heroku にデプロイすることにより実行することができる．

(1) 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

本プログラムのエラー処理を以下に示す．

(1) (機能 2) のリクエストメッセージに存在しない都道府県名があった場合，Slack の発言を受信したチャンネルに対し，結果が得られなかったという内容の発言を行う．

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 POST リクエストを受け取ったときの動作

(2) 表 1 に示す動作環境以外でプログラムを実行

参考文献

[1] SUNTORY HOLDINGS: BAR-NAVI（バーナビ） Web API, SUNTORY（オンライン），入手先⟨http://webapi.suntory.co.jp/barnavidocs/⟩（参照2016-04-19）.