

SlackBot プログラムの仕様書

2016 年 04 月 21 日
乃村研究室 坪川 友輝

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 3 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 “ ” と発言する機能
- (2) “weather” という発言に反応して、岡山市の天気情報を発言する機能
- (3) GitHub へ push を行ったとき、push したことを発言する機能

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) GitHub アカウント

3 機能

本プログラムがもつ 3 つの機能を以下に述べる。本プログラムは Slack に対するユーザの発言を受信し、受信した内容に対応する内容を Slack へ発言する。ただし、受信する発言は “@T-Bot” で始まる発言のみである。

(機能 1) “「 」と言って” という文字列を含む発言に反応して “ ” と発言する機能
この機能は、受信した発言の中に “「 」と言って” という文字列が含まれる場合、“ ” と発言する機能である。

(機能 2) “weather” という発言に反応して、岡山市の天気情報を発言する機能
この機能は、受信した発言の “@T-Bot” に続く文字列が “weather” である場合、岡山市の天気情報を発言する機能である。なお、天気情報は Web サイト [1] から取得する。発言する内容は天気概況文、3 日間の天気予報、および予想気温である。なお、予想気温は取得した情報に含まれている場合のみ発言する。

表 1: 動作環境

項目	内容
ソフトウェア	Ruby(2.1.5),bundler(1.11.2),GitHub(2.1.4),Heroku(3.42.47)
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

(機能3) GitHub へ push を行ったとき，push したことを発言する機能
この機能は，Webhooks を設定したリポジトリへ push を行った時，push を行ったユーザ，および最新の commit の内容を発言する機能である．

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．

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 の WebHooks の設定
- (2) GitHub の WebHooks の設定
- (3) Heroku 上にアプリケーションを生成

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 の WebHooks の設定

- (1) 自身の Slack アカウントにログインする．

- (2) 以下の URL から「APP Directory」にアクセスし、「Configure」をクリックする。
<https://slack.com/apps>
- (3) 「Custom Integrations」から「Incoming WebHooks」をクリックする。
- (4) 「Add Configuration」をクリックし、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した後、「Add Incoming WebHooks integration」をクリックし、Webhook URL を取得する。
- (6) 再び「Custom Integrations」にアクセスし、「Outgoing WebHooks」をクリックする。
- (7) 「Add Configuration」から、「Add Outgoing WebHooks integration」をクリックする。
- (8) Outgoing WebHook に関して以下を設定する。
 - (A) 発言を監視する channel
 - (B) WebHooks が動作する契機となる Trigger Word (@T-Bot)
 - (C) WebHooks が動作した際に POST を行う URL
https://<app_name>.herokuapp.com/git

5.2.2 GitHub の WebHooks の設定

- (1) 自身の GitHub アカウントにログインする。
- (2) 任意のリポジトリの管理画面にアクセスする。
- (3) 「Webhooks & services」から「Add webhook」をクリックする。
- (4) push された時に POST を行う URL を設定して、「Add webhook」をクリックし新しい Webhooks を作成する。

5.2.3 Heroku 上にアプリケーションを生成

- (1) 以下の URL から 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) Heroku から送信されたメールに記載されている URL をクリックし , パスワードを設定する .
- (3) 登録したアカウントでログインし , 「Getting Started with Heroku」の使用する言語を選択する .
- (4) 「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 .
- (5) 以下のコマンドを実行し , Heroku にログインする .

```
$ heroku login
```

- (6) 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 <app_name>
```

- (7) 以下のコマンドを実行し , Incoming WebHook URL を Heroku の環境変数に追加する . XXXXXXXXX には Slack の Webhooks 設定時に取得したものを入力する .

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

- (8) 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

本プログラムを実行するための手順を以下に示す .

- (1) コマンドラインに以下のコマンドを入力し , Heroku にアプリケーションをデプロイすることで実行する .

```
$ 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) 設定した OutgoingWebhooks 以外から POST された場合，エラーメッセージを表示して発言を中止する．

ERROR: POST is delivered by wrong whooks.

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) 受信した発言中に “「 」と言って” が複数含まれるときの Slack への発言
- (2) GitHub の Webhooks の設定で push 以外のイベントでも POST を行うようにした場合
- (3) “@T-Bot” 以外の文字列を Trigger Word として Bot を動作させた場合
- (4) 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] livedoor Co.,Ltd.: Weather Hacks - livedoor 天気情報, livedoor Co.,Ltd. (オンライン), 入手先
< http://weather.livedoor.com/weather_hacks > (参照 2016-4-21) .