SlackBot プログラムの仕様書

2020/6/2

野村 優文

1 概要

本資料は，2020 年度新人研修課題で作成した SlackBot プログラムの仕様をまとめたものである．本プログラムではチャットツールである Slack[1] を用いる．また，SlackBot は，ユーザが Slack 上で投稿した特定の文章をきっかけとして，Slack 上で自動的に返信する機能をもつ．本プログラムは以下の２つの機能をもつ．

(1) ユーザから指定された文字列を返信する機能
(2) Weather Hacks（気象データ配信サービス）[2] を用いた天気予報の情報を返信する機能

2 対象とする利用者

本プログラムでは，Slack アカウントを所有する利用者を対象としている．

3 機能

ユーザが Slack 上に “@masabot ” で始まる文章を投稿したとき，本プログラムは受け取った文字列に対応する文章を Slack 上に返信する．本プログラムが返信する内容は, ユーザが投稿する “@masabot ” の後ろに付く文字列によって決まる．以下に本プログラムがもつ機能について記述する．

(機能 1) ユーザから指定された文字列を返信する機能

この機能は，ユーザが “@masabot 「(指定した文字列)」と言って” と投稿したとき，SlackBot プログラムは “「” と “」と言って” の間にある文字列を読み込んで， “(指定した文字列)” と返信する．例えば，ユーザは “@masabot 「こんにちは」と言って” と投稿したとき．Slackbot は “こんにちは” と返信する．ただし， “(指定した文字列)” の中に “「” もしくは “」” を使用してはならない．

(機能 2) Weather Hacks（気象データ配信サービス）を用いて天気予報の情報を返信する機能

この機能は， Weather Hacks（気象データ配信サービス）を用いて天気予報の情報を返信する．ユーザは， “@masabot (日にち) の (都道府県名) の天気” と Slack 上に投稿したとき，SlackBot は指定した日にちと都道府県の天気予報を返信する．例えば，ユーザが “@masabot 明日の東京の天気” と投稿したとき，SlackBot は図 1 のように返信する．この機能を使用するにあたって，ユーザが指定できる “(日にち)” には， “今日”， “明日”， “明後日” の 3 種類のうちのいずれかを入力する．“(都道府県名)” には， “東京” や “岡山” など，日本の都道府県名を 1 つ入力する．

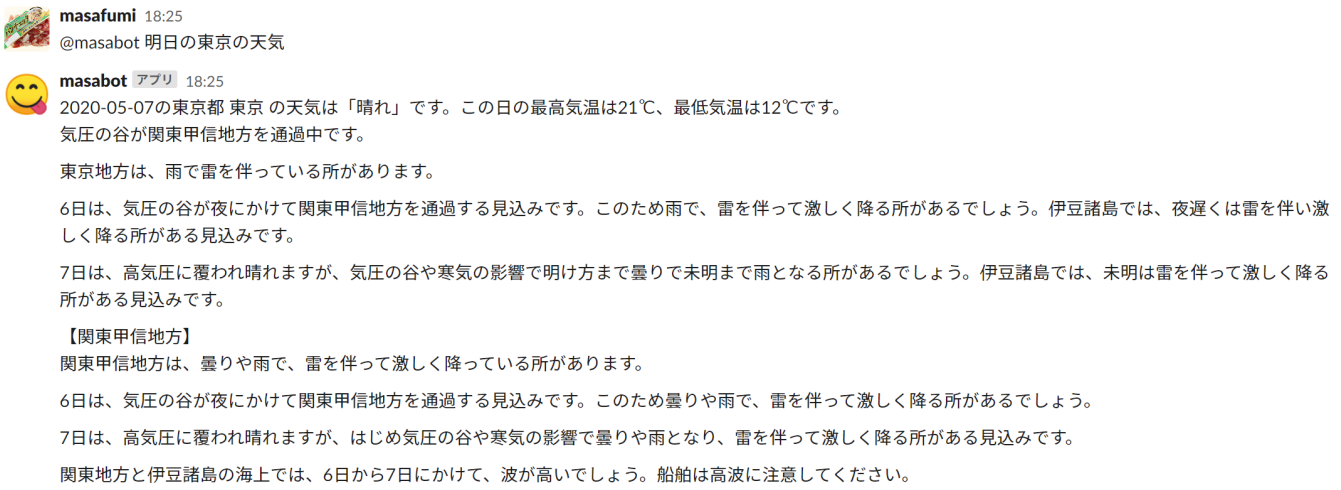

図 1　ユーザが “@masabot 明日の東京の天気” と投稿したときの SlackBot の返信

また，“北海道” は “道” という文字を含めて “(都道府県名)” に入力する．

ユーザが (機能 1)，(機能 2) を使用するために必要な文字列以外の文章を，Slack に投稿したとき，本プログラムは以下の文章を返信する．

こんにちは、masabot です。
"@masabot 「○○○」と言って"と投稿した場合、"○○○"と返信します。
天気予報を知りたいときは、"@masabot □□の△△の天気"
(□□は「今日」「明日」「明後日」のいずれか)(△△は都道府県名) と投稿してください。
この日の指定した都道府県の天気を返信します。

4　動作環境

本プログラムの動作環境を表 1 に示す．

5　環境構築

本プログラムを実装するための環境の構築手順を以下に記述する．

(1) Slack の Incoming WebHook の設定
(2) Slack の Outgoing WebHook の設定
(3) Heroku アカウントの作成と設定

上記の手順の詳細の構築方法を以下に記述する．

表 1　動作環境

項目	内容
OS	Ubuntu 18.04.2 LTS
CPU	Intel(R) COre(TM) m3-6Y30 CPU @ 0.90GHz
メモリ	4.0GB
Ruby	ruby 2.5.5p157
Gem	activesupport 6.0.2.2
	bundler 2.1.4
	json 2.1.0
	openssl 2.1.2
	rack 2.2.2, 2.0.4
	rack-protection 2.0.8.1, 2.0.1
	sinatra 2.0.8.1, 2.0.1
	tilt 2.0.10, 2.0.8

5.1　Slack の Incoming WebHook の設定

Slack における Incoming WebHook の設定を以下の手順で行う．

(1) 自分の Slack アカウントにログインする．
(2) Slack の左上のメニューを開き，「その他管理項目」の「アプリを管理する」を開く．
(3) カスタムインテグレーションから，「Incoming WebHook」を開く．
(4)「Slack に追加」した後，Incoming WebHook がメーセージを投稿するチャンネルを選択し，インテグレーションを追加する．
(5) 名前をカスタマイズする．また，Webhook URL を 5.3 節 (9) で使用するので控えておく．
(6)「設定を保存する」を選択する．

5.2　Slack の Outgoing WebHook の設定

Slack における Outgoing WebHook の設定を以下の手順で行う．

(1) 自分の Slack アカウントにログインする．
(2) Slack の左上のメニューを開き，「その他管理項目」から「アプリを管理する」を開く．
(3) カスタムインテグレーションから，「Outgoing WebHook」を開く．
(4)「インテグレーションを追加する」を選択する．
(5) Outgoing WebHook に関して，以下の設定を行う．

(A)「チャンネル」にて，ユーザが投稿した特定の文字列を監視するチャンネルを設定する．
(B)「引き金となる言葉」にて，WebHook が動作する契機となる文字列を入力する．
(C)「名前をカスタマイズ」にて，設定する Outgoing WebHook の名前を入力する．

(6)「設定を保存する」を選択する．

5.3 Heroku アカウントの作成と設定

Heroku アカウントの作成と設定を以下の手順で行う．

(1) 以下の URL より Heroku にアクセスし，「新規登録」より Heroku アカウントを登録する．
https://jp.heroku.com/

(2) 登録したアカウントでログインし，「Getting Started on Heroku」にて，「Ruby」を選択する．

(3) 以下のコマンドを実行し，Heroku CLI をインストールする．

```
$ sudo snap install heroku --classic
```

(4) heroku コマンドを使用するために，以下のコマンドを実行し，パスを通す．

```
$ export PATH=$PATH:/snap/bin/
```

(5) 以下のコマンドを実行し，Heroku CLI がインストールできていることを確認する．

```
$ heroku version
```

(6) 以下のコマンドを実行し，Heroku CLI にログインする．

```
$ heroku login
```

(7) 作成したアプリケーション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>
```

なお，アプリケーション名は小文字と数字，およびハイフンのみ使用できる．

(8) 以下のコマンドで，生成した Heroku のアプリケーションが，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)
```

(9) 以下のコマンドを実行し，5.1 節 (5) で控えておいた Incoming WebHook URL を，Heroku の環境変数に追加する．

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

6　使用方法

本プログラムを使用するには，以下の手順で行う．

(1) 以下に示す Git リポジトリから本プログラムを取得する．
`git@github.com/masafumi0612/BootCamp.git`
(2) 以下のコマンドを用いて，本プログラムを Heroku にデプロイする．
`git push heroku master`

7　エラー処理

(機能 2) を使用する際に，投稿する文字列の中に (日にち) が含まれていないときや，使用できる “今日”，“明日”，“明後日” 以外の文字列を使用したとき，“今日” の指定した都道府県の天気予報を返信する．例えば，“東京の天気” や “明々後日の東京の天気” という文字列を投稿した際は，今日の東京の天気予報を返信する．また，投稿する文字列の中に都道府県名が含まれていないとき，本プログラムはの以下にように返信する．

都道府県名を入力してください。
天気予報を知りたいときは、"@masabot □□の△△の天気"
（□□は「今日」「明日」「明後日」のいずれか）（△△は都道府県名） と投稿してください。
指定した日にちの都道府県の天気を返信します。

これに当てはまる例として，ユーザが “@masabot 明日の天気” という文字列を投稿したときが挙げられる．

8　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記述する．

(1) 4 章に示した動作環境以外の環境で，本プログラムを実行したときの動作
(2) 本プログラムが，Slack の Outgoing WebHook 以外から POST リクエストを受け取って実行した際の動作

参考文献

[1] Slack: その仕事、Slack で。，Slack Technologies.Inc（オンライン），入手先 ⟨https://slack.com/intl/ja-jp/⟩（参照 2020-05-15）．

[2] Hacks, W.: Weather Hacks（気象データ配信サービス），日本気象協会（オンライン），入手先 ⟨http://weather.livedoor.com/weather_hacks/⟩（参照 2020-05-15）.