SlackBot プログラムの仕様書

2017 年 4 月 20 日
乃村研究室 山本 瑛治

1 概要

本資料は，平成 29 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 つの機能をもち，Slack[1] アカウントを所有する利用者を対象としている．

(1) “「○○○」と言って” という文字列を含む発言に反応して “○○○” と返信する機能

(2) “○月△日は何の日” もしくは “今日は何の日” という文字列を含む発言に反応して, 該当する日付に関する情報を返信する機能

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 機能

本プログラムは，“@YBot ” という文字列から始まる発言を受信し，返信を行う．また，“@YBot ” 以降の内容によって Bot の返信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 “○○○” と返信する機能

Bot の受信した発言が “「○○○」と言って” という文字列を含む場合，“○○○” と返信する．また “「「○○○」と言って」と言って” のように返信する文字列の解釈が複数存在する場合は，最長の文字列を返信する．つまりこの場合は “「○○○」と言って” と返信する．

(機能 2) “○月△日は何の日” もしくは “今日は何の日” という文字列を含む発言に反応して，該当する日付に関する情報を返信する機能

Bot の受信した発言が “○月△日は何の日” もしくは “今日は何の日” という文字列を含む場合，該当する日付に関する Wikipedia の記念日・年中行事の情報

を返信する．“○月△日は何の日” という文字列の場合，日付の指定には半角数字を用いる必要がある．

また Bot の受信した発言が “今日は何の日” という文字列を含む場合，リクエストを受けとった日付に関する情報を返信する．この際の日付は Slack の time zone を基に決定される．

なお Bot が受信した発言内に “○月△日は何の日” もしくは “今日は何の日” という文字列が複数存在する場合，以下の順序で本機能が反応する文字列が判定される．

(1) 位置にかかわらず “今日は何の日” という文字列が存在する場合，この文字列に反応し，返信を行う．

(2) 最初に存在する文字列に反応し，返信を行う．
例：“4 月 1 日は何の日 5 月 2 日は何の日” という文字列を受信した場合，4 月 1 日に関する情報を返信する．

Bot の返信内容としては，以下の情報が含まれる．

(1) 検索対象となる日付

(2) 日付の情報の総件数

(3) 日付の情報の表示件数
情報の総件数が５件以下の場合は全件表示し，５件より多い場合は検索結果の上位５件のみ表示する．

(4) 日付に関する情報
Wikipadia の日付ページから情報を取得する．

(5) リンク
Wikipadia の日付ページへのリンクが記載される．

例えば “@YBot 4 月 14 日は何の日” という発言に対する Bot の返信は，図 1 のようになる．

なお，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，“Hello” という文字列を返信する．

また (機能１)，(機能 2) の条件が重複する場合は (機能 1) の条件が優先される．つまり “「今日は何の日」と言って” という発言の場合は “今日は何の日” という返信を行う．

図 1: “4 月 14 日は何の日” に対する Bot の返信例

3 動作環境

本プログラムの動作環境を表 1 に示す.

4 動作確認済み環境

動作確認済み環境は表 1 の環境である.

5 使用方法

5.1 環境構築

5.1.1 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) Heroku のアカウント作成と設定

(4) Gem のインストール

表 1: 動作環境

項目	内容
OS	Linux Debian GNU/Linux(version 8.1)
CPU	Intel(R) Core(TM) i5-4670 CPU (3.40GHz)
メモリ	8.00GB
ブラウザ	FireFox バージョン 52.0.2
ソフトウェア	Ruby バージョン 2.1.5
	bundler バージョン 1.14.6
	heroku CLI バージョン 5.7.16
	Git バージョン 2.1.4

以後の項では具体的な手順を述べる.

なお本プログラムで利用した API は Wikipadia の情報を取得できる MediaWiki API[2] であり，この API は登録等を行わずに誰でも利用できる.

5.1.2 Slack アカウントの Incoming WebHooks の設定

Slack が提供する Incoming WebHooks の設定を行う．Incoming WebHooks とは指定された URL へ POST することで，POST された情報を利用した発言をすることができる機能である.

(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「Manage」をクリックする.

(4) ページ左側の「Custom Integrations」をクリックする.

(5) 「Incoming WebHooks」をクリックする.

(6) 「Add configuration」をクリックする.

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

(8) Webhook URL を取得する.

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)	@YBot
URL(s)	https://<myapp_name>.herokuapp.com/slack myapp_name は 5.1.4 項 (10) にて指定するアプリケーション名

5.1.3 Slack アカウントの Outgoing WebHooks の設定

Slack が提供する Outgoing WebHooks の設定を行う．Outgoing WebHooks とは指定された発言をすることで，指定した URL にその発言の情報を POST する機能である．

(1) 5.1.2 項の (4) まで同様の手順を踏む.

(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」をクリックする.

(5) 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.

(6) 「Save Settings」をクリックする.

5.1.4 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 for free」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 on 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 the Heroku CLI for…」右側の下矢印から対応OSを選択する．

(7) 「Download the Heroku CLI for…」をクリックして表示されるコマンドを用いてCLI(Command Line Interface)のダウンロードを行う．

(8) 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，CLIがインストールされたことを確認する．

```
$ heroku version
```

(9) 以下のコマンドを実行し，Herokuにログインする．

```
$ heroku login
```

(10) 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．

(11) 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)
```

(12) 以下のコマンドを実行し，Herokuの環境変数に5.1.2項(8)で取得したWebhook URLを設定する．

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

5.1.5 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
```

5.2 実行手順

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
```

6 エラー処理と保証しない動作

6.1 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) において不正な日付に関する情報を取得しようとした場合は，“Hello” という文字列を返信する．

6.2 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 以外の POST リクエストをブロックする動作

(2) 表 1 に示す動作環境以外でプログラムを実行

(3) Wikipedia の該当日付のページの記念日・年中行事の項目が削除された場合の動作

参考文献

[1] Slack: Slack: Where work happens, Slack (online), available from ⟨https://slack.com/⟩ (accessed 2017-04-17).

[2] Media Wiki: API:メインページ，Media Wiki（オンライン），入手先 ⟨https://www.mediawiki.org/wiki/API:Main_page/ja⟩（参照 2017-04-17）.