SlackBot プログラムの仕様書

2019/4/26

松尾 和樹

1　はじめに

本資料は，2019 年度 B4 新人研修課題で作成した 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 のチャンネルに自動で投稿を行ったり，設定された契機となる単語によってユーザへの返信を自動で行ったりするものである．本プログラムは以下の 2 つの機能を持つ．

(1) ユーザから指定された文字列を投稿する機能
(2) ユーザとしりとりを行う機能

本資料において，投稿内容は引用符 “” で囲って示す．

2　対象となる利用者

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) Yahoo! JAPAN ID

Yahoo! JAPAN ID は API キーの取得に使用する．

3　SlackBot プログラムの処理の流れ

SlackBot プログラムの処理の流れを図 1 に示す． 図 1 中の (1) から (7) の処理を以下に示す．

(1) ユーザから Slack サーバに投稿する
(2) Slack サーバは，ユーザからの投稿を契機に SlackBot サーバに POST する
(3) SlackBot サーバは Slack サーバからの POST を処理する
(4) SlackBot サーバは API にリクエストを送信する
(5) リクエストを受け取った API は SlackBot サーバにレスポンスを返却する
(6) SlackBot サーバは API から受け取ったレスポンスを処理する
(7) SlackBot サーバから Slack サーバへ投稿する
(8) Slack サーバは，ユーザに SlackBot サーバからの投稿内容を表示する

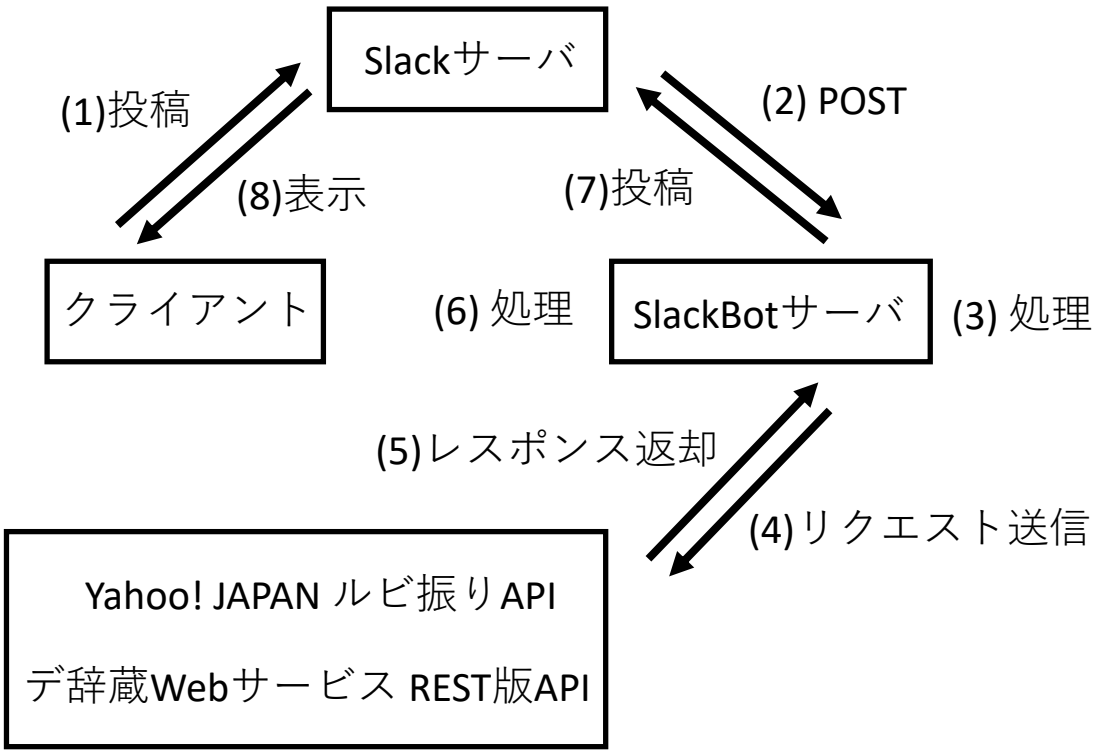

図 1　SlackBot プログラムの処理の流れ

4　機能

4.1　概要

本プログラムは，Slack での “@matsubot” から始まる投稿を受信し，この投稿に対して返信を行う．返信内容は “@matsubot” に続く文字列によって決定される．

4.2　コマンドの説明

本プログラムに実装されているコマンド一覧を表 1 に示す．コマンドは “`@matsubot <command>`” のように使用する．`<command>`は表 1 に示すコマンドを指定する．各コマンドについて以下で説明する．

(1) start コマンド
start コマンドは以下のメッセージを投稿し，しりとりを開始した状態へ遷移する．

しりとりを開始します．
入力を行ってください．

(2) end コマンド

end コマンドは以下のメッセージとともに，しりとりを終了した状態へ遷移する．

しりとりを終了します．

(3) status コマンド

status コマンドはしりとり中か否かを表示する．しりとり中であれば以下のように投稿する．

現在しりとりの途中です．初めの文字は"（文字）"です．

初めの文字とはユーザが投稿するべき単語の初めの文字である．たとえば，SlackBot が直前に投稿した単語の最後の文字が「あ」であれば以下のように投稿する．

現在しりとりの途中です．初めの文字は"あ"です．

また，しりとり中でなければ以下のように投稿する．

現在しりとりは行なっていません．

(4) help コマンド

help コマンドはヘルプメッセージを表示する．投稿するヘルプメッセージの内容を以下に示す．

こんにちは．
matsubot です．
機能の説明をします．
1．「○○と言って」と言われると「○○」と返します．
2．しりとりを行うことができます．しりとりの細かいルールを説明します．
(1) 使用可能な文字はひらがな，カタカナ，漢字のみです
(2)「ー」は無視します　例：「ルビー」の最後の文字は「ビ」です
(3) 拗音の小文字は大文字として扱います　例：「バッシュ」の最後の文字は「ユ」です

コマンドは以下の通りです．
help:このヘルプメッセージを表示
start:しりとりの開始
end:しりとりの終了
status:しりとり中かそうでないかを表示

4.3 機能の説明

以下で本プログラムが持つ 2 つの機能を述べる．

(機能 1) ユーザから指定された文字列を投稿する機能

ユーザの “@matsubot （文字列）と言って” という投稿に対して（文字列）の部分を返信する．たとえば，“@matsubot こんにちはと言って” という投稿に対しては “こんにちは” という文字列を返信する．

表 1　SlackBot プログラムのコマンド一覧

コマンド名	説明
start	しりとりを開始する.
end	しりとりを終了する.
status	しりとり中か否かを表示する.
help	ヘルプメッセージを表示する.

(機能 2)　ユーザとしりとりを行う機能

ユーザは “@matsubot （単語）” のように投稿する．SlackBot はユーザから受け取った単語の最後の文字から始まる単語を返信する．しりとりの処理の手順を以下に示す．ただし，“@matsubot start” と投稿し，しりとりを開始している状態とする.

(1) SlackBot は受け取った単語を Yahoo! JAPAN のルビ振り API[2] を用いてひらがなへ変換する

(2) 変換後の文字列の最後の文字から始まる単語のリストをデ辞蔵 Web サービス REST 版 API[3] の検索メソッドを使用して取得する
デ辞蔵 Web サービス REST 版 API の辞書には Edict 和英辞典を指定する.

(3) 取得した単語のリストからランダムで 1 件選択する

(4) 選択した単語をデ辞蔵 Web サービス REST 版 API の内容取得メソッドにより単語の品詞と英訳を取得する

(5) 単語の品詞が名詞であれば，その単語をユーザへ返信する
名詞でない場合は，(3) から再度行う.

上記の（機能 1）と（機能 2）どちらにも当てはまるメッセージを受信した場合は（機能 1）が優先される.

また，上記の（機能 1）と（機能 2）どちらの条件にも当てはまらないメッセージを受信した場合，SlackBot は以下のように投稿する.

登録されていないコマンドです.
`@matsubot help` でヘルプメッセージを表示します.

5　動作環境

本プログラムは Heroku 上で動作する．Heroku とは，PaaS で，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．Heroku の環境を表 2 に示す.

表 2　動作環境（Heroku）

項目	内容
OS	Ubuntu 18.04.2 LTS (Bionic Beaver)
CPU	Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz
メモリ	512MB
Ruby	2.5.3p105
Gem	bundler 1.16.1
	mini_portile2 2.4.0
	mustermann 1.0.3
	nokogiri 1.6
	rack 2.0.7
	rack-protection 2.0.5
	sinatra 2.0.1
	tilt 2.0.9

6　環境構築

6.1　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) Yahoo! JAPAN Web サービスのアプリケーション ID 発行
(2) Incoming Webhook の設定
(3) Outgoing Webhook の設定
(4) Heroku の設定

次節で各項目の具体的な手順を述べる．

6.2　具体的な手順

6.2.1　Yahoo! JAPAN Web サービスのアプリケーション ID 発行

Yahoo! JAPAN デベロッパーネットワーク [4] の Web API を利用するにはアプリケーション ID の発行が必須である．以下の手順でアプリケーション ID を発行する．

(1) 以下の URL にアクセスし，アプリケーションの登録を行う．

```
https://e.developer.yahoo.co.jp/register
```

表 3　Outgoing Webhook の設定項目

項目	内容
チャンネル	投稿を監視する channel
引き金となる言葉	Webhook が動作する契機となる単語
URL	Webhook が動作した際に POST を行う URL

(2) アプリケーションの登録後，アプリケーション ID が表示される．

6.2.2　Incoming Webhook の設定

Incoming Webhook の設定を行う．Incoming Webhook とは，外部サービスから指定した Slack のチャンネルにメッセージを投稿する機能である．以下に設定手順を示す．

(1) Slack にログインした状態で以下の URL にアクセスする．
`https://<team_name>.slack.com/apps/manage/custom-integrations`
ここで，`<team_name>`は自分の所属するワークスペースに変更する．ワークスペースとは，他のメンバとコミュニケーションをとり業務を行うための共有スペースである．
(2)「Incoming Webhook」をクリックする．
(3)「設定を追加」から，新たな Incoming Webhook を追加する．
(4)「チャンネルへの投稿」の項目からメッセージを送信するチャンネルを選択する．
(5)「設定を保存する」をクリックし，Webhook URL を取得する．

6.2.3　Outgoing Webhook の設定

Outgoing Webhook の設定を行う．Outgoing Webhook とは指定されたチャンネルに指定された条件の投稿が行われた際に事前に設定した URL にリクエストを POST する機能である．以下に設定手順を示す．

(1) Slack にログインした状態で以下の URL にアクセスする．
`https://<team_name>.slack.com/apps/manage/custom-integrations`
ここで，`<team_name>`は自分の所属するワークスペースに変更する．
(2)「Outgoing Webhook」をクリックする．
(3)「設定を追加」から，新たな Outgoing Webhook を追加する．
(4) Outgoing Webhook の設定を行う．設定する項目を表 3 に示す．本プログラムでは，引き金となる言葉は “@matsubot”，URL は Heroku で作成したアプリケーションの URL である．

6.2.4 Heroku の設定

Heroku の設定を行う．以下に Heroku の設定手順を示す．

(1) `https://www.heroku.com/`へアクセスし，「Sign up」から新しいアカウントを登録する．
(2) 登録したアカウントでログインし，「Getting Started on Heroku」の使用する言語として Ruby を選択する．
(3)「I’m ready to start」をクリックする．「Download the Heroku CLI for…」から OS を選択し，Heroku CLI 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
```

インストールが成功していれば，例として以下のような出力が得られる．

```
heroku/7.22.9 darwin-x64 node-v11.10.1
```

(5) 以下のコマンドを実行し，Heroku にログインする．

```
$ heroku login
```

(6) 作成した SlackBot 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 <myapp_name>
```

ただし，`<myapp_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，Heroku の環境変数に Incomig Webhook URL を追加する．

```
$ heroku config:set INCOMING_WEBHOOK_URL=<Incomig Webhook URL>
```

ここで，`<Incomig Webhook URL>`は 6.2.2 項の (5) で取得した Webhook URL に変更する．
(8) 以下のコマンドで Heroku の環境変数にアプリケーション ID を追加する．

```
heroku config:set YAHOO_API_KEY=<Application ID>
```

ここで，`<Application ID>`は 6.2.1 項の (2) で表示されたものに変更する．

7 使用方法

本プログラムは，GitHub で管理されている．本プログラムのリポジトリを以下に示す．

`https://github.com/matsuo0227/BootCamp.git`

本プログラムは，上記 BootCamp リポジトリ内の `2019/SlackBot/MySlackBot.rb` である．

本プログラムは Heroku 上で動作するため，Heroku にデプロイして使用する．SlackBot プログラムを配置しているディレクトリで以下のコマンド実行する．

```
$ git push heroku master
```

8 エラー処理

本プログラムのエラー処理を以下に示す．

(1) Yahoo! JAPAN ルビ振り API は，ユーザからの投稿に API の仕様上使用できない漢字が含まれている場合，エラーを返す．Yahoo! JAPAN ルビ振り API が上記のエラーを返した場合，以下のように投稿する．
その文字は認識不可能です！

9 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 Webhook 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: Where work happens, Slack (online), available from ⟨https://slack.com/intl/ja-jp/⟩ (accessed 2019-04-25).

[2] ヤフー株式会社：Yahoo! JAPAN ルビ振り API，ヤフー株式会社（オンライン），入手先 ⟨`https://developer.yahoo.co.jp/webapi/jlp/furigana/v1/furigana.html`⟩（参照 2019-04-25）．

[3] イースト株式会社：デ辞蔵 Web サービス REST 版 API，イースト株式会社（オンライン），入手先 ⟨`https://www.est.co.jp/dev/dict/rest/`⟩（参照 2019-04-25）．

[4] ヤフー株式会社：Yahoo! JAPAN デベロッパーネットワーク，ヤフー株式会社（オンライン），入手先 ⟨`https://developer.yahoo.co.jp/`⟩（参照 2019-04-25）．