

# SlackBot プログラム仕様書

2018/4/26

吉田 修太郎

## 1 概要

## 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持する利用者を対象としている。

- (1) Slack アカウント
- (2) Heroku アカウント
- (3) GitHub アカウント

GitHub アカウントは、本プログラムによって、issue の追加する機能、issue の一覧を投稿する機能を利用する場合に必要となる。

## 3 機能

本プログラムは、Slack での “@SYBot” から始まる投稿に対して、その投稿の “@SYBot” 以降の内容に応じて投稿する。本プログラムが持つ機能について、以下に述べる。

### (機能 1) 任意の文字列を返信する機能

ユーザが “@SYBot 「(任意の文字列)」と言って” と投稿した場合、その投稿中の (任意の文字列) の部分のみを投稿する。たとえば、ユーザから “@SYBot 「はじめまして」と言って” という投稿があれば、“はじめまして” と投稿する。

### (機能 2) 設定された GitHub リポジトリの issue を一覧にして返信する機能

ユーザが “@SYBot get issue” と投稿した場合、設定された GitHub リポジトリの open 状態の issue を取得し、取得した issue 全てのタイトルを返信する。

### (機能 3) 設定された GitHub リポジトリに issue を追加する機能

ユーザが “@SYBot make issue t:(title)b:(body)” と投稿した場合、設定された GitHub リポジトリに新たな issue を追加する。投稿中の (title) と (body) は任意の文字列である。ただし、(title) は作成する issue のタイトルであり、(body) は作成する issue の本文である。“t:(title)” の部分は入力必須であるが、“b:(body)” の部分は入力しなくてもよい。さらに、issue の追加が正常に行われた場合は “created” を投稿し、追加に失敗した場合は “creation fail:(message)” を投稿する。投稿中の (message) は、GitHub の API から返されるエラーの種類を示す文字列である。

## 4 動作環境

本プログラムは Heroku 上で動作する。Heroku とは、ソフトウェアを構築し、稼働させるためのプラットフォームである。Heroku の動作環境を表 1 に示す。

表 1 Heroku の動作環境

OS	Debian 8.1
CPU	Intel(R) Core(TM) i5-4590
メモリ	1.0GB
Ruby	ruby 2.1.5p273
Ruby Gem	bundler 1.16.1 mustermann 1.0.2 rack 2.0.4 rack-protection 2.0.1 tilt 2.0.8 sinatra 2.0.1

## 5 環境構築

### 5.1 概要

本章では、本プログラムの動作のために、環境構築を行う。環境構築のために設定が必要な項目を以下に示す。

- (1) Heroku の設定
- (2) Incoming WebHooks の設定
- (3) Outgoing WebHooks の設定
- (4) 環境変数の設定

上記の各項目について、次節で項を設け説明する。

### 5.2 手順

#### 5.2.1 Heroku の設定

- (1) Heroku アカountの作成

以下の URL にアクセスし、「Sign up」から新規の Heroku アカountを作成する。

<https://www.heroku.com/>

(2) 使用する言語の登録

登録したアカウントで Heroku にログインし、「Getting Started with Heroku」の使用する言語として「Ruby」を選択する。

(3) CLI のダウンロード

「I'm ready to start」をクリックし、「Download Heroku CLI for...」から CLI をダウンロードする。CLI とは Command Line Interface のことであり、ダウンロード後に端末から Heroku のコマンドが実行可能となる。

(4) アプリケーションの作成

Heroku にログインし、Heroku 上にアプリケーションを作成する。以下のコマンドを実行し、ログインする。

```
$ heroku login
```

ログイン後、本プログラムの存在するディレクトリで以下のコマンドを実行する。

```
$ heroku create <app_name>
```

ただし、<app\_name>は作成するアプリケーション名である。アプリケーション名はアルファベット小文字、数字、およびハイフンのみからなる任意の文字列である。

### 5.2.2 Incoming WebHooks の設定

(A) 以下の URL にアクセスする。

```
https://<team_name>.slack.com/apps/manage/custom-integrations
```

ただし、<team\_name>は自分の Slack アカウントの所属チーム名である。

(B) 「Incoming WebHooks」をクリックする。

(C) 「Add Configuration」をクリックする。

(D) 「Choose a channel...」から本プログラムから受信した文字列を投稿するチャンネルを選択する。

(E) 「Add Incoming WebHooks integration」をクリックする。

(F) 「Webhook URL」の欄に表示されている URL を控えておく。

(G) 「Save Settings」をクリックする。

### 5.2.3 Outgoing WebHooks の設定

(A) 以下の URL にアクセスする。

```
https://<team_name>.slack.com/apps/manage/custom-integrations
```

ただし、<team\_name>は自分の Slack アカウントの所属チーム名である。

(B) 「Outgoing WebHooks」をクリックする。

(C) 「Add Configuration」をクリックする。

(D) 「Add Incoming WebHooks integration」をクリックする。

(E) 各項目を設定する .

- (a) 「Channel」の項目に , 本プログラムに投稿を監視させたいチャンネルを設定する .
- (b) 「Trigger Word(s)」の項目に , “@SYBot” を設定する .
- (c) 「URL(s)」の項目に , `https://<app_name>.herokuapp.com/slack` を設定する .  
ただし , <app\_name>は , 5.2.1 項で作成したアプリケーション名である .
- (d)

(F) 「Save Settings」をクリックする .

#### 5.2.4 環境変数の設定

以下のコマンドを実行し , Heroku の環境変数を設定する .

```
$ heroku config:set INCOMING_WEBHOOK_URL="<incoming_webhook_url>"  
$ heroku config:set USERNAME=<username>  
$ heroku config:set PASSWORD=<password>
```

ただし , <incoming\_webhook\_url>は 5.2.2 項で控えておいた URL である . 実行後 , Incoming WebHooks の URL , GitHub の Username , および GitHub の Password が環境変数として設定される .

## 6 使用方法

本プログラムは , Heroku にデプロイして使用する . 本プログラムを Heroku で管理しているディレクトリで , 以下のコマンドを実行する .

```
$ git push heroku master
```

実行後 , 本プログラムが Heroku にデプロイされ , 本プログラムの機能が利用可能となる .

## 7 エラー処理と保証しない動作

### 7.1 エラー処理

### 7.2 保証しない動作

## 参考文献

- [1] 乃村研究室 資料 <https://github.com/nomlab/BootCamp/blob/master/2018/README.org>