

2018年度 Web勉強会

2018年4月3日

乃村研究室

小倉 伊織, 西 良太, 山本 瑛治

概要

<目的>

Webの仕組みや, SlackBotを作成するために必要な技術の理解

<目次>

- (1) Webの仕組み(HTTP)
- (2) WebHooksを利用したSlackとの通信
- (3) APIを用いた外部のプログラムの利用

Web

<Webとは>

インターネット上に存在するサーバとの通信によって、コンテンツのやりとりを行うシステム

<Webの仕組み>

(1) インターネット上にWebのコンテンツ(Webページ)を保持するWebサーバが点在していて、一意な識別子によって管理



URI(Uniform Resource Identifier)

(2) Webサーバに対して、クライアントがコンテンツの取得や追加を要求



HTTP(Hypertext Transfer Protocol)

(3) コンテンツをユーザに理解しやすい形で表示し、コンテンツからコンテンツに自由に遷移可能



HTML(HyperText Markup Language)

HTTP

<HTTPとは>

- (1) TCP/IPベースの通信のプロトコル
- (2) クライアントとサーバによる通信
 - (A) クライアントからサーバにリクエストを送信
 - (B) サーバはリクエストに応じて処理
 - (C) 処理結果をレスポンスとしてクライアントに返却
- (3) ステートレスな通信
 - サーバがクライアントの状態を保持しない

<HTTPにおける主なリクエスト>

GET : サーバからデータを取得

POST : サーバへデータを追加

Webサーバとブラウザの関係

例) <http://www.okayama-u.ac.jp/index.html> にアクセス

通信プロトコル: HTTP

通信先(サーバ): www.okayama-u.ac.jp

サーバへの要求: /index.html を GET する

Webブラウザ
(クライアント)



(1) リクエスト送信



Webサーバ



(3) レスポンス返却



(4) 返却内容を整形して表示

(2) 要求に対応した処理

index.html を返却

Webhooks

Webサービスで発生する何らかのイベントを契機に、外部のWebサービスに通知する仕組み

例：URLへのPOST
Slackのコメント

< SlackでのWebhooksの活用 >

Incoming WebHooks : 指定したURLに特定の文字列がPOSTされたとき、Slackにその文字列を送信

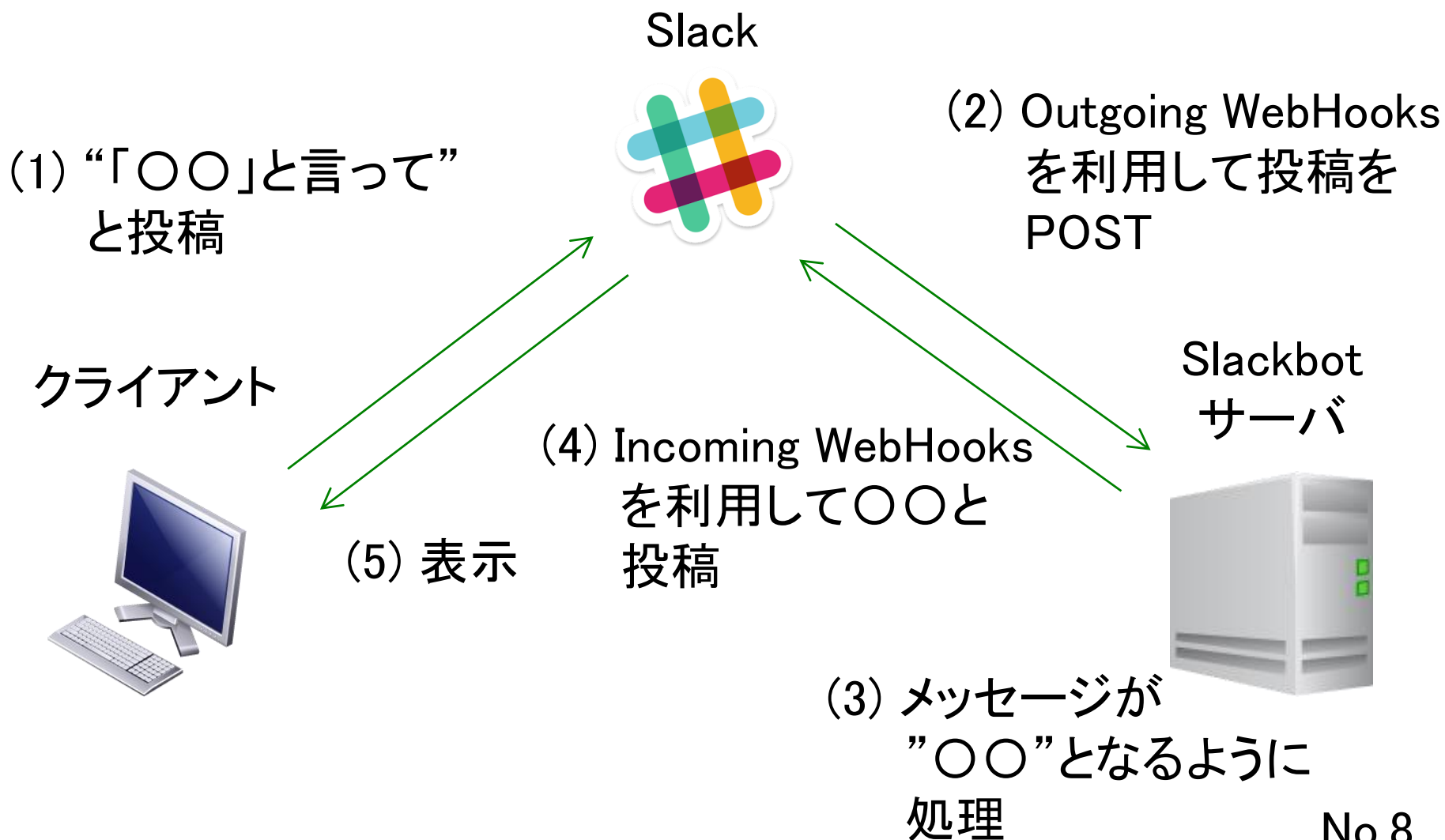
Outgoing WebHooks : Slackで特定の文字列が投稿されたとき、指定したURLにPOSTする

作成したアプリケーションが動作している場所

- (1) Webサービスから見えるグローバルな場所
- (2) HTTP通信に対応

WebHooksを利用したSlackとの通信

例：“「〇〇」と言って”の投稿に対して〇〇と投稿する



Heroku



- (1) 作成したアプリケーションを
クラウド上にデプロイするサービス
- (2) Gitによる容易なデプロイが可能

グローバルな場所にデプロイできるため、
Outgoing WebHookが利用可能

WebAPI

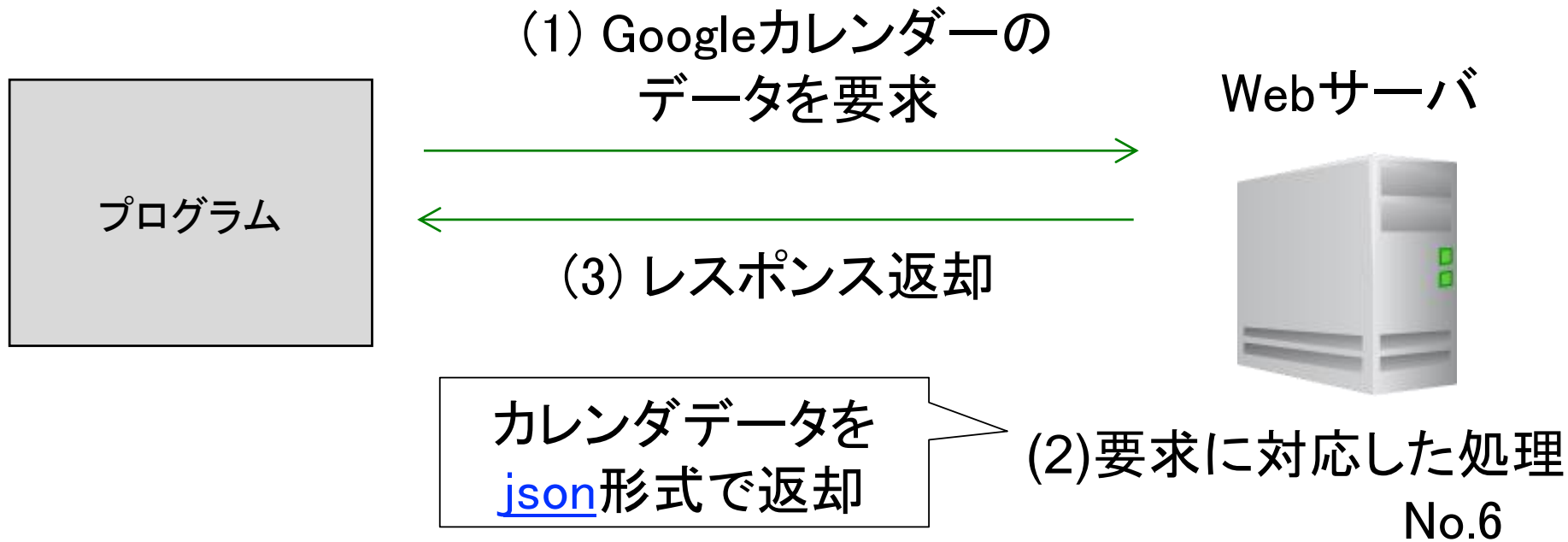
<API(Application Programming Interface)とは>

あるプログラム内の関数を
別のプログラムから使えるようにするためのインタフェース

<WebAPIとは>

HTTPの通信を用いたAPI

例：Googleカレンダー, twitter, Facebook など



APIを利用したSlackBotの仕組み

