

セットアップ準備: Google Drive 内のソース設定

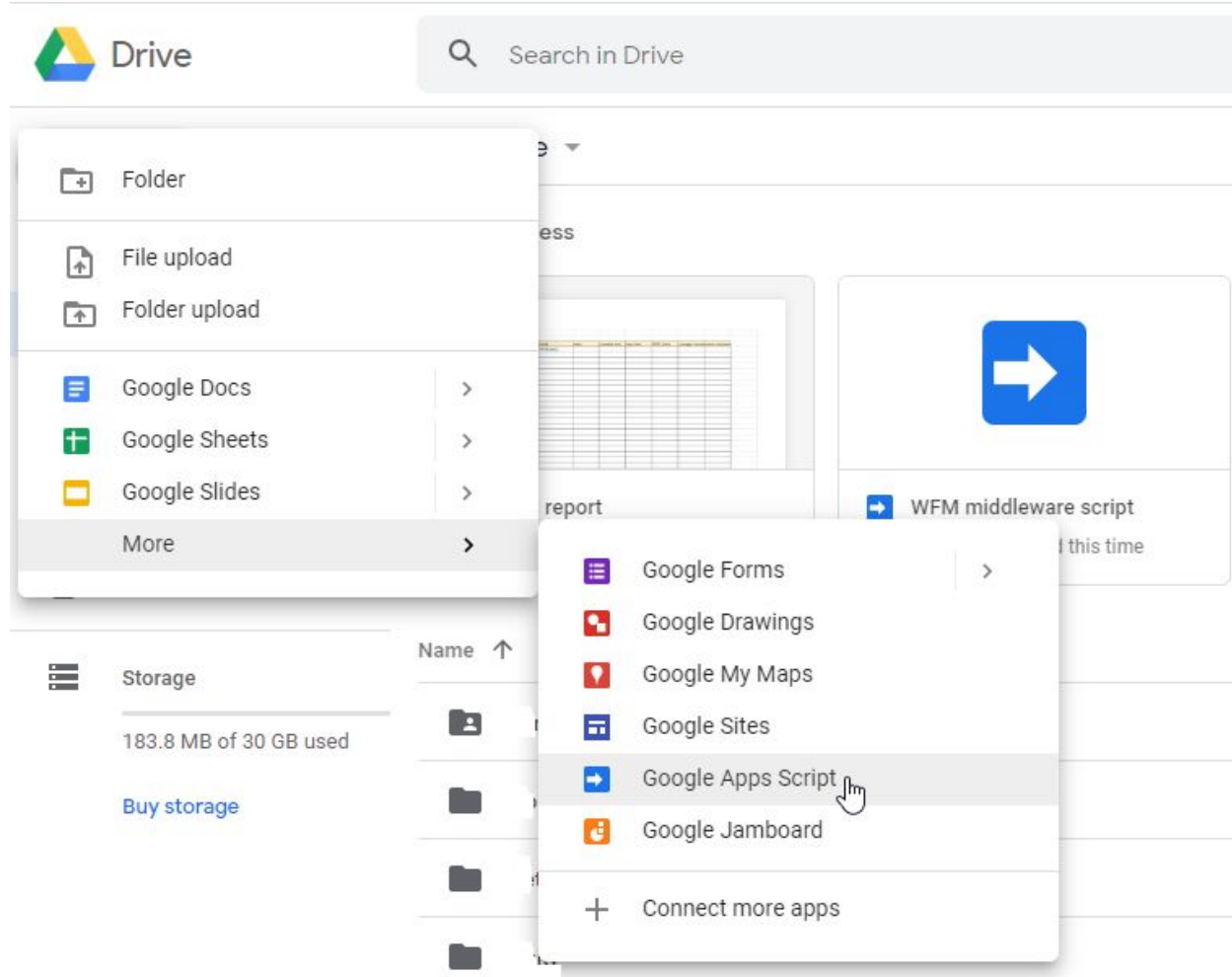
1. Google Driveに新規フォルダを作成

<https://drive.google.com/> にアクセスし、「WFH」というフォルダ名で新規フォルダを作成します。

2. Google Apps Script を新規作成

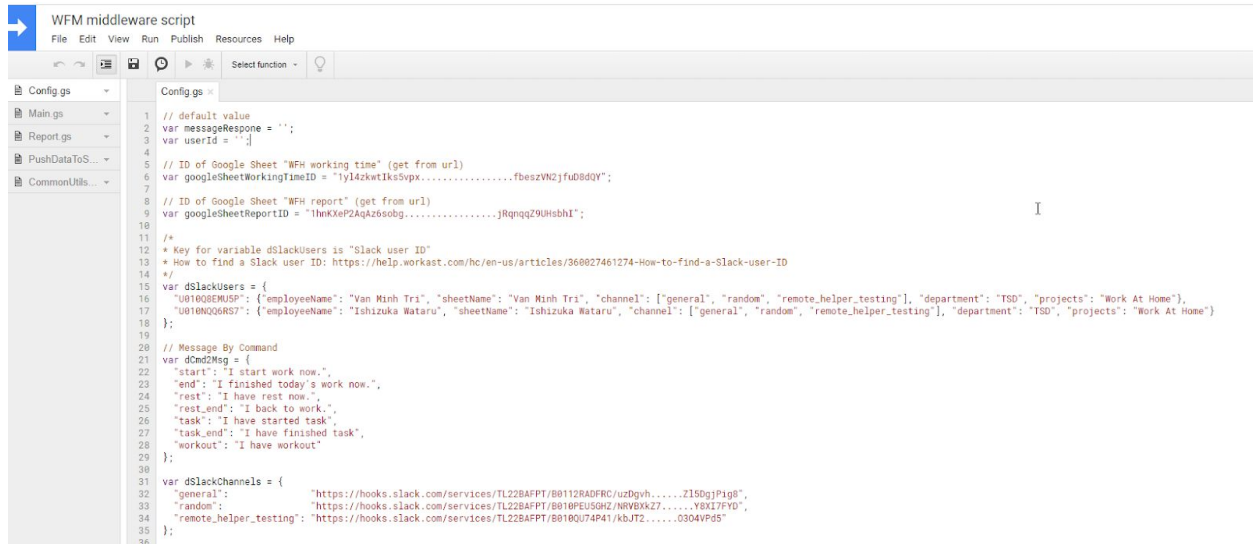
「WFH」フォルダ内に、Google Apps Script を新規作成します。Google Apps Script の項目が見つからない場合は、次のように設定します。

「新規」 → 「その他」 → 「アプリを追加」で表示されるポップアップウィンドウ内で、「Google Apps Script」を検索



a) 新規作成された Google Apps Script を開きます。

Apps Script Editor（Google Apps の設定を調整したり、スクリプトを書き換えたりする画面）が表示されます。



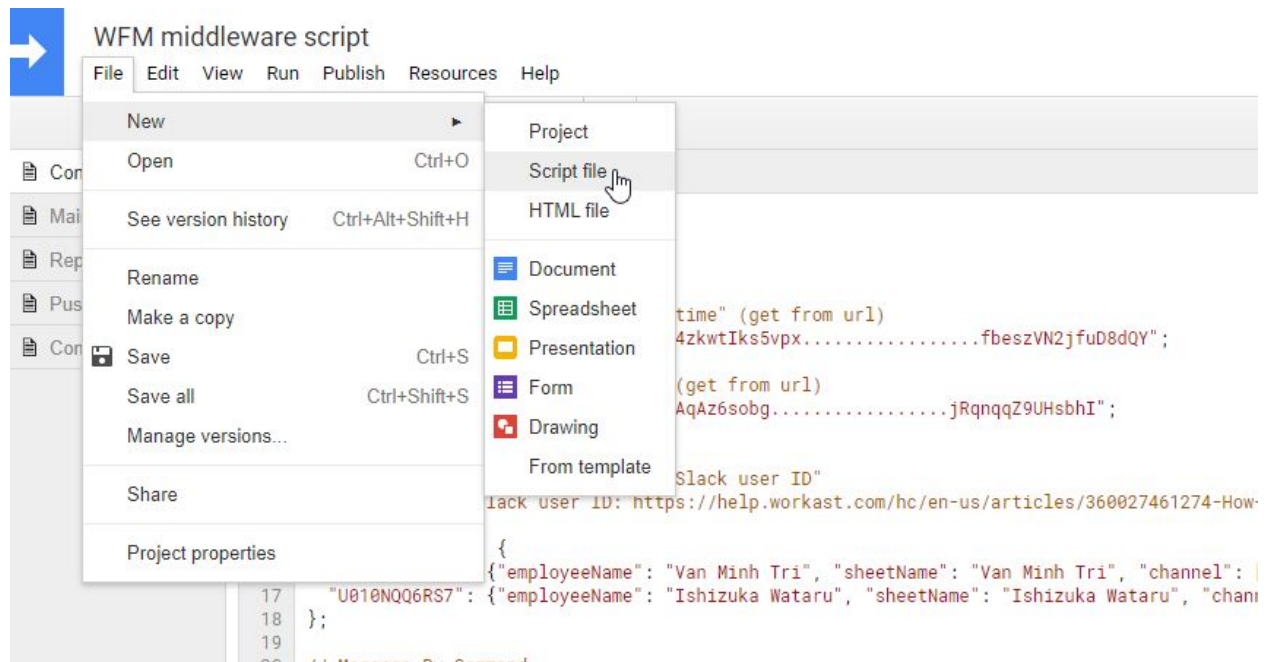
```
1 // default value
2 var messageResponse = '';
3 var userId = '';
4
5 // ID of Google Sheet "WFM working time" (get from url)
6 var googleSheetWorkingTimeID = "1y14zket1ks5vpx.....fbeszvW2jfu08dQY";
7
8 // ID of Google Sheet "WFM report" (get from url)
9 var googleSheetReportID = "1hkX6P2Aq4z6sobg.....JRqmqQZ9U4HbhlI";
10
11 /*
12 * Key for variable dSlackUsers is "Slack user ID"
13 * How to find a Slack user ID: https://help.workast.com/hc/en-us/articles/360827461274-How-to-find-a-Slack-user-ID
14 */
15 var dSlackUsers = {
16   "0816080EUSP": { "employeeName": "Van Minh Tri", "sheetName": "Van Minh Tri", "channel": ["general", "random", "remote_helper_testing"], "department": "TSD", "projects": "Work At Home" },
17   "0816080EUSP": { "employeeName": "Ishizuka Wataru", "sheetName": "Ishizuka Wataru", "channel": ["general", "random", "remote_helper_testing"], "department": "TSD", "projects": "Work At Home" }
18 };
19
20 // Message By Command
21 var dCmdMsg = {
22   "start": "I start work now.",
23   "end": "I finished today's work now.",
24   "rest": "I have rest now.",
25   "rest_end": "I back to work.",
26   "task": "I have started task.",
27   "task_end": "I have finished task.",
28   "workout": "I have workout"
29 };
30
31 var dSlackChannels = {
32   "general": "https://hooks.slack.com/services/TL22BAFPT/80112RADFRG/uzDgvh.....Z150gjP1g8",
33   "random": "https://hooks.slack.com/services/TL22BAFPT/8010PEUSGHZ/HRVEXKZ7.....Y0X17PpYD",
34   "remote_helper_testing": "https://hooks.slack.com/services/TL22BAFPT/8018QU74P41/kbJT2.....0304VPd5"
35 };
36
```

b) Apps Script Editor にて、5つのスクリプトファイルを作成します。

画面上部のツールバーから、「ファイル」→「New」から「スクリプト ファイル」を選択することで、同一のGoogle Apps Script 内にスクリプトファイルを新規作成できます。これを5回つ作成します。

それぞれのファイル名とコードは下記よりコピー＆ペーストしてください。

<https://github.com/grevo-vn/Work-From-Home/tree/master/GoogleAppsScript>

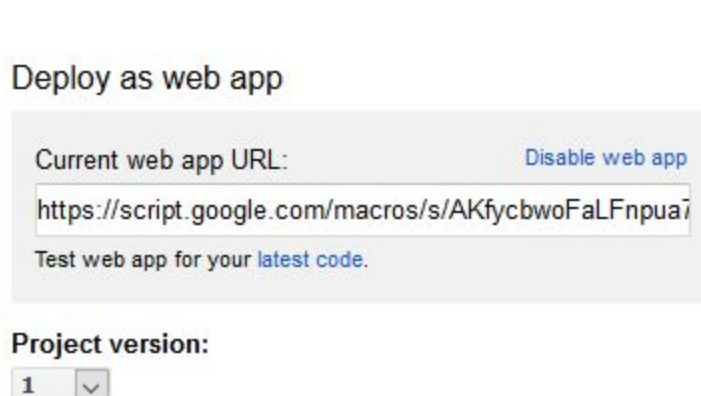


c) 完成したスクリプトをウェブアプリとして公開します。

上部ツールバーから「公開」 → 「ウェブアプリケーションとして導入...」を選択します。

アプリの公開範囲を設定するポップアップが開くので、「アプリケーションにアクセスできるユーザ」を「全員(匿名)」にします。

「導入」をクリックすることで、ウェブアプリとしての利用準備が完了します。



導入完了後に表示される「現在のウェブアプリケーションのURL」は、後ほど Slack Apps の設定で必要になるので、コピーやメモしておきましょう。

※GAS（Google Apps Script）を利用する理由について

Slackからの入力を受信・データをスプレッドシートに保管・出力し、Slackへの投稿をすることが目的です。これらに必要なサーバとしての役割、アプリとしての役割を一手に担うことができるため、広く利用されているGoogle Accountとの親和性も高いGASを利用しています。

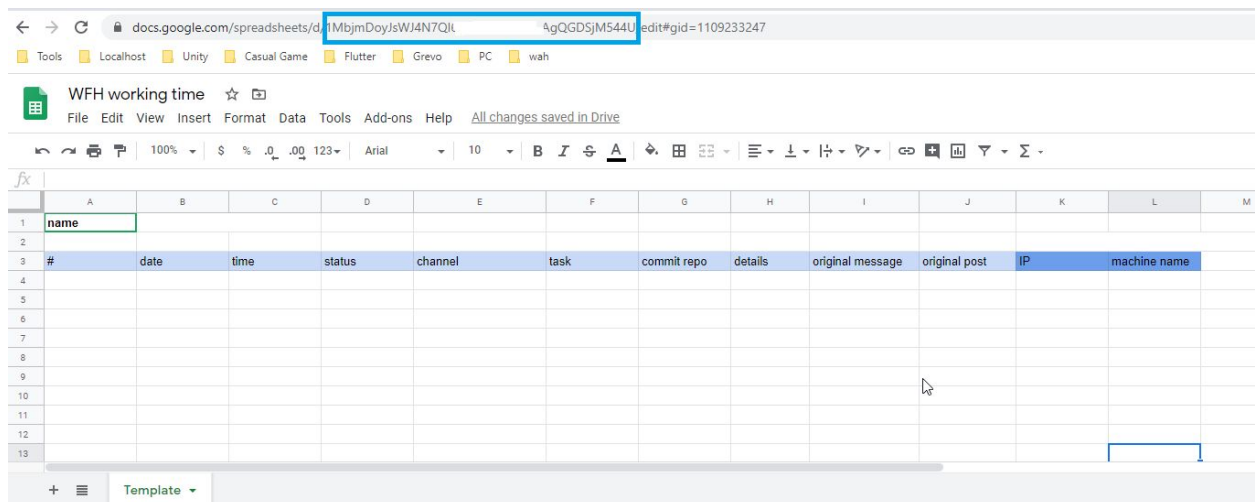
スクリプトを改修したい場合は、GASの開発者向け情報をご参照ください。

<https://developers.google.com/apps-script>

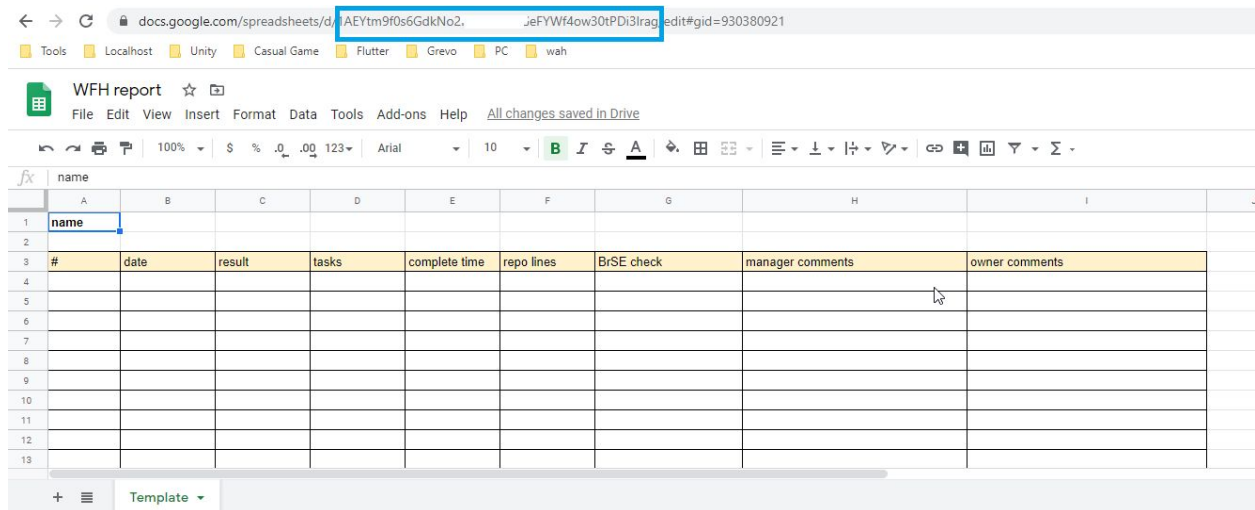
3. スプレッドシートを新規作成

「WFH」フォルダ内に2種類のスプレッドシートを作成します。

a) WFH working time



b) WFH report



Notice:

新たなメンバーが追加される際、シートを複製して使う前提のため、一番左のシート名を「Template」にします。

以上でセットアップ前の準備は完了となります。

このままではアプリは機能しませんが、後の工程で再びGASに戻ってスクリプトを編集します。その際、ここで作成したスプレッドシート2種類のID（URL上の d/ と /edit の間の文字列）が必要になります。

Slack Apps セットアップ

1. Slack Apps を新規作成する

Web版SlackのSlack Apps設定ページ (https://api.slack.com/apps?new_app=1) にアクセスし、必要な情報を埋めていきます。

「App Name」 にアプリ名を入れ、「Development Slack Workspac」 " にはアプリを利用したいワークスペースを選択します。最後に "Create App" をクリックすれば、Slack 内で稼働できるアプリが作成されます。

以降の工程で、アプリをインストールしたり、機能やコマンドを追加していきます。

Create a Slack App



App Name

e.g. Super Service

Don't worry; you'll be able to change this later.

Development Slack Workspace

Development Slack Workspace



Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel

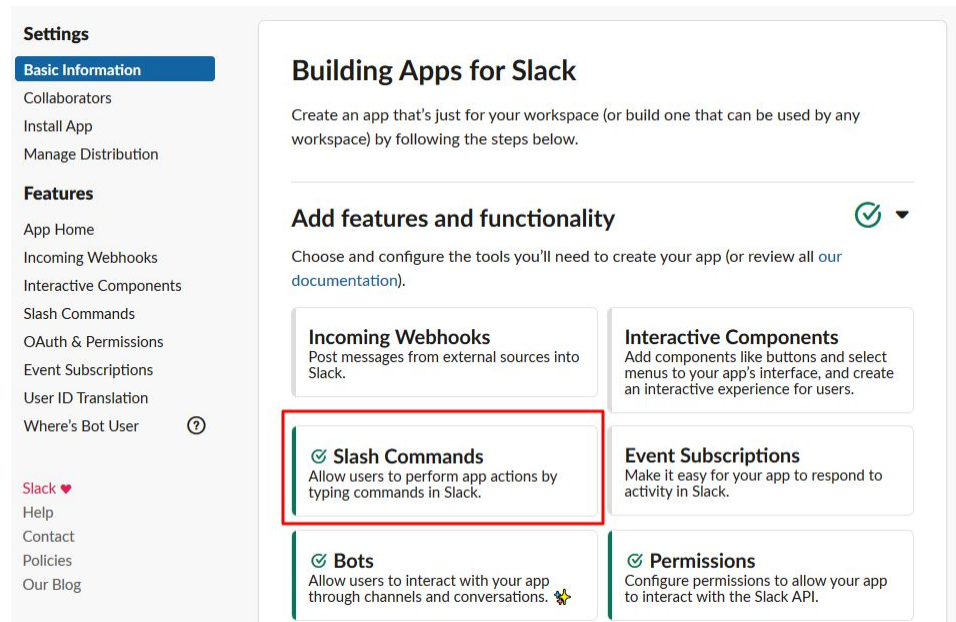
Create App

※ Slack Apps の詳細については、公式ページ (<https://api.slack.com/start/overview>) から確認できます。

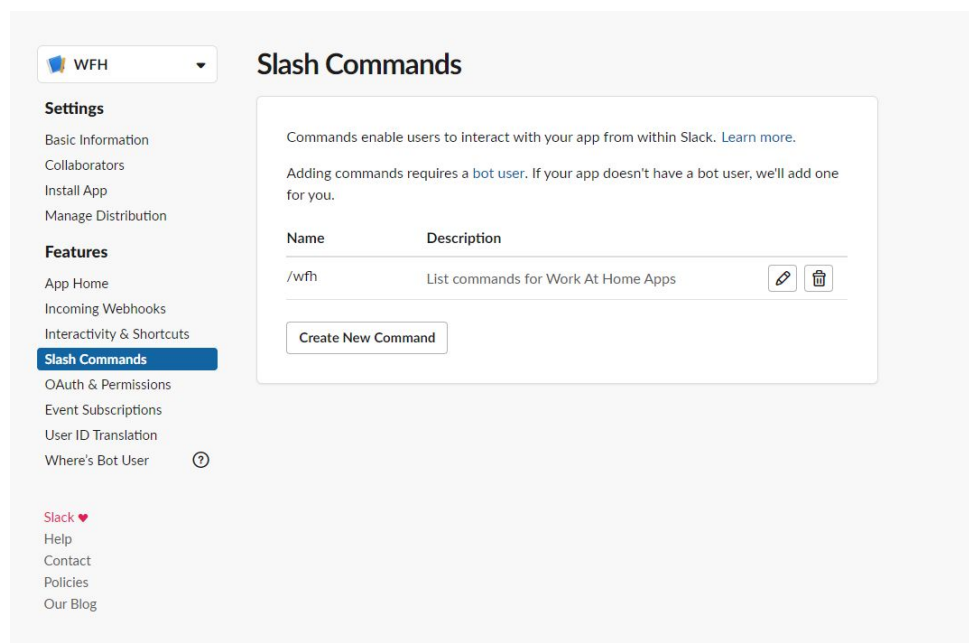
2. /コマンドを設定する

Slack上でのコマンド入力によって、アプリが動作するように設定します。

a) 「Add features and functionality」の項目内にある「Slash Commands」をクリックします。



b) 「Create New Command」をクリックし、それぞれコマンドを作成します。



c) コマンド設定に必要な情報を入力します。

Command	<input type="text" value="/wfh"/>
Request URL	<input type="text" value="https://script.google.com/macros/s/..."/>
Short Description	<input type="text" value="List commands for Work At Home Apps"/>
Usage Hint	<input type="text" value="start -msg [string] rest -msg [string] re..."/> <small>Optionally list any parameters that can be passed.</small>
Escape channels, users, and links sent to your app	
Unescaped: @user #general	
Preview of Autocomplete Entry	
<div><div>Commands matching "wfh"</div><div>WFH</div><div>/wfh start -msg [string] rest -... List commands for Work At ...</div></div> <div><div>+</div><div>/wfh</div></div>	

Command

「/wfh」と入力します。（どのコマンドで起動するかを設定）

Request URL

[セットアップ準備](#)の項目で新規作成した Google Apps Script の「現在のウェブアプリケーションのURL」を入力します。

Short Description

アプリの説明です。Slack 上で他のユーザーの目にも触れるので、分かりやすい内容が推奨されます。

Usage Hint

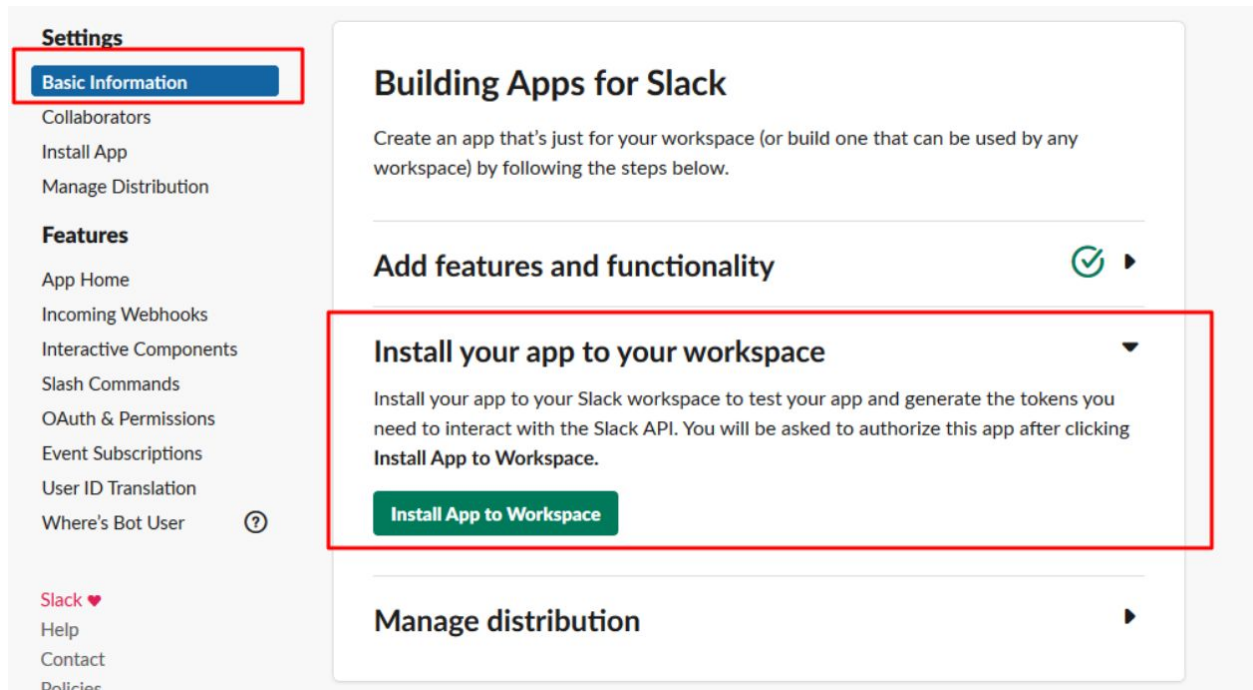
その他使えるコマンドがあれば、ユーザーガイドとして記載できます。
本アプリでは、作業の開始/休憩/終了時など、特定のタイミングに文字列を投稿できる下記のコマンドが使用可能です。

- ・ start -msg [文字列]
- ・ rest -msg [文字列]
- ・ rest_end -msg [文字列]
- ・ task [ID] -msg [文字列]
- ・ task_end [ID] -msg [文字列]
- ・ end -msg [文字列]

最後に「Save」をクリックすれば、コマンドの設定は完了です。

3. Slack Apps をワークスペースに追加する

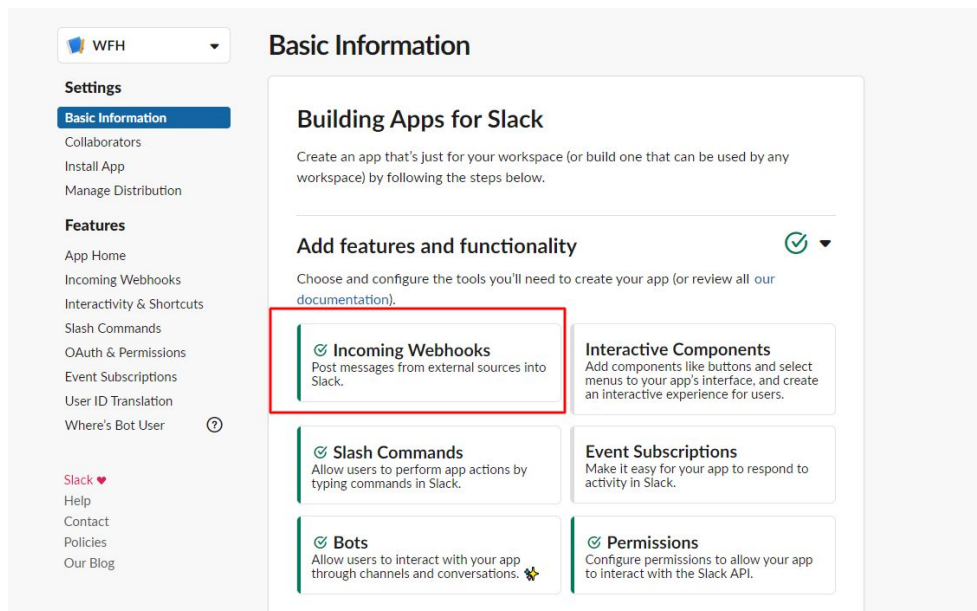
「Install your app to your workspace」の項目内にある「Install App to Workspace」をクリックします。これで現在のワークスペースにアプリが追加されます。



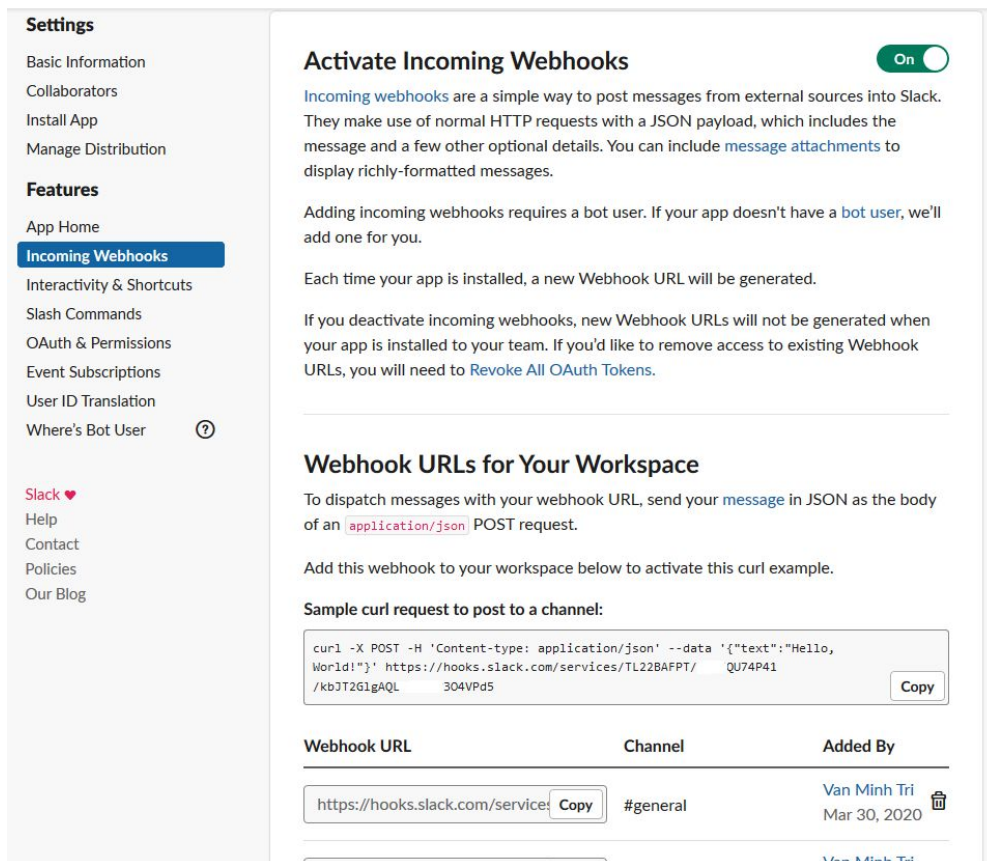
4. WebHook を登録する

文言を各チャンネルに投稿するために、Webhook（特定のコマンド実行などによって、指定されたURLにデータを追記する仕組み）を設定します。

a) 「Add features and functionality」にある「Incoming Webhooks」をクリック。



b) 右上の「Active Incoming Webhooks」を On にします。



c) 「Add new Webhook to Workspace」をクリックして、Slackのどのチャンネル上でアプリを稼働させるか設定します。

Webhook URLs for Your Workspace




To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5
```

Copy

Webhook URL	Channel	Added By
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#general	Van Minh Tri Mar 30, 2020 
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#random	Van Minh Tri Mar 30, 2020 
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#remote_helper_testing	Van Minh Tri Mar 30, 2020 
Add New Webhook to Workspace		

d) 作業状況を投稿したいチャンネルを選択し、「Allow」をクリックします。

wah is requesting permission to access the GREVO Co., Ltd. Slack workspace



Where should wah post?

wah requires a channel to post to as an app

A dropdown menu showing the selected channel for posting. The text "# general" is visible, and a small downward arrow is on the right side of the box.

Cancel

Allow

以上でSlack Appsでの設定は完了です。次に、アプリの機能を編集します。

Config.gs セットアップ

Google Apps Script を再度開き、Slack Apps で設定したコマンドが有効に機能するように、Webhookとスプレッドシートを紐づけます。

編集するのは「Config.gs」になります。



1. スプレッドシートIDを指定する

以下の変数に、[セットアップ準備](#)で新規作成した対応するスプレッドシートのIDを代入します。（IDはスプレッドシートURLの ~d/...../edit で挟まれた箇所 の文字列）

```
var googleSheetWorkingTimeID = "WFH working time シートのID"
```

```
var googleSheetReportID = "WFH report シートのID"
```

2. 利用ユーザー情報を入力する

変数 dSlackUsers に、アプリを利用するユーザー情報を入力します。

※アプリの動作不良を避けるために、JavaScriptまたはGAS習得者による入力を推奨。

必要な情報としては、①SlackユーザーID、②利用者名、③利用チャンネル名（複数可）の3種類になります。

var dSlackUsers = {...} のカッコ内に、以下のフォーマットで入力します。

```
"SlackのユーザーID": {"userName": "利用者名", "sheetName": "利用者名", "channel": ["利用チャンネル1", "利用チャンネル2", "利用チャンネル3"]}
```

同じカッコ内で、カンマで区切って同じフォーマットを追加すれば、複数メンバー分の情報を追記できます。

```
var dSlackUsers = {  
  
  "U010Q8EMU5P": {"userName": "Van Minh Tri", "sheetName": "Van Minh Tri", "channel": ["general", "random", "remote_helper_testing"]},  
  
  "U000X0XXX0X": {"userName": "Owlcat", "sheetName": "Owlcat", "channel": ["general", "random", "remote_helper_testing"]},  
  
  ...  
};
```

① SlackユーザーIDの取得

Slackのアカウントプロフィールから確認できます。

② userName

ユーザー名を入力します。

sheetName

ここで入力した値がスプレッドシート上で自動的に出力されます。

③ channel

コマンドを入力した際に投稿が行われるチャンネルを指定します。

3. Webhook URLの反映

変数 dSlackChannels にアプリを動かすチャンネルとWebhookの情報を入力します。
※アプリの動作不良を避けるために、JavaScriptまたはGAS習得者による入力を推奨。

必要な情報は①Channel名、②Webhook URLで、いずれも Slack Apps (<https://api.slack.com/apps>) の設定画面で確認できます。
設定画面はメニュー内「Features」にある「Incoming Webhook」で表示できます。

Webhook URLs for Your Workspace




To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5
```

Copy

Webhook URL	Channel	Added By
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#general	Van Minh Tri Mar 30, 2020 
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#random	Van Minh Tri Mar 30, 2020 
https://hooks.slack.com/services/TL22BAFPT/B7U74P41/kbJT2G1gAQL0304VPd5 Copy	#remote_helper_testing	Van Minh Tri Mar 30, 2020 
Add New Webhook to Workspace		

設定画面に記載された情報は、

var dSlackChannels = {...} のカッコ内に、以下のフォーマットで入力します。

```
"Channel名": "WebhookのURL"
```

同じカッコ内で、カンマで区切って同じフォーマットを追加すれば、複数チャンネル分の情報を追記できます。

```
var dSlackChannels = {  
  "general": "https://hooks.slack.com/services/ABC...jklm1",  
  "random": "https://hooks.slack.com/services/DEF...nopq2",  
  "remote_helper_testing":  
    "https://hooks.slack.com/services/GHI...rstu3",  
  ...  
};
```

諸注意

スクリプトを編集した際は、再度公開するまでその編集内容は反映されません。

セットアップ準備の時と同じく、ツールバーから「公開」→「ウェブアプリケーションとして導入...」と選択していきます。

ポップアップ内で「プロジェクトバージョン」を「新規作成」にし、「更新」をクリックすることで内容が反映されます。

×

Deploy as web app

Current web app URL:

[Disable web app](#)

`https://script.google.com/macros/s/AKfycbwtSQdPKjlmC`

Test web app for your [latest code](#).

Project version:

New ▾

Describe what has changed...

Execute the app as:

Me (minhtri.van@grevo.net) ▾

You need to authorize the script before distributing the URL.

Who has access to the app:

Anyone, even anonymous ▾

Update

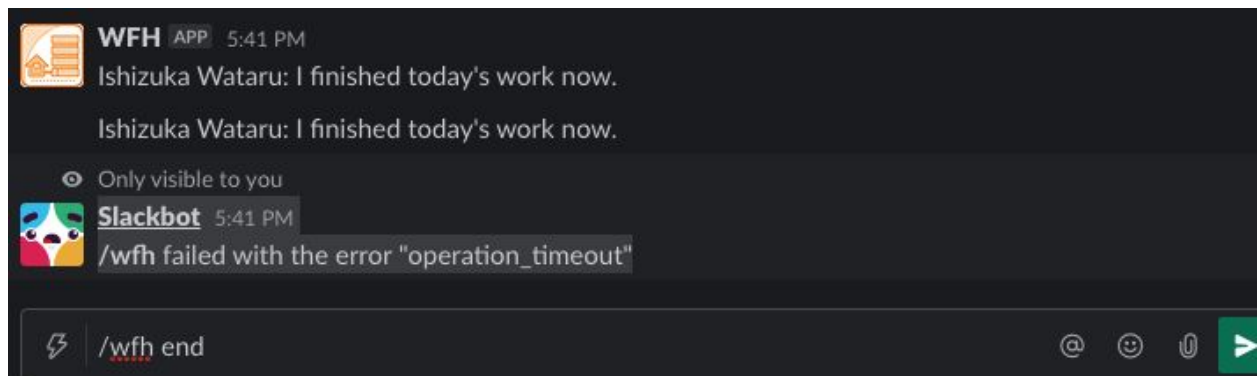
Cancel

[Help](#)

また、実際にSlack上で /wfh コマンドを実行した際、operation_timeout というエラーログが返ってくることがあります。

これはSlackの仕様によるもので、コマンド実行から3秒が経過すると、実行に失敗したと内部で判定され、Slackbotが自動的にメッセージを送信するために発生します。

実際のスクリプト上は問題なくコマンドが機能しているので、WFH APP の動作に影響はありません。



エラーが表示されていても、上記画像のようにコマンドの実行結果が正しく表示されていれば、問題ありません。