

Cómo construir un bot para Slack

@davsket

[https://github.com/davsket/slack-bot-
experiment](https://github.com/davsket/slack-bot-experiment)

David
Avellaneda

@davsket

Co-organizador de:

BogotáJS

JSConf.co

Char.la

ColombiaDev

Co-fundador de:

Monoku





- Soy Tauro
- Me gusta correr en la playa
- Tengo un hijo (el de la foto)
- En serio me gusta mucho correr
- Me gusta compartir lo que se

Cómo construir un Servidor y Bot de Memes para Slack

@davsket



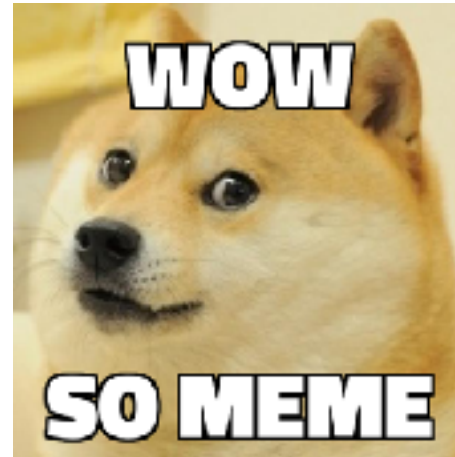
Agenda

Agenda

- ¿Qué vamos a hacer? v2
- ¿Qué necesitamos? v2
- JUST DO IT!!! step++ v2
- Momento philosoraptor v2



¿Qué vamos a hacer?



¿Qué necesitamos?

¿Qué necesitamos?

- promesas (yup, de nuevo)
- como funciona slack (APIs)
- CommonJS (require)
- variables de entorno
- node / express (básico) *
- request *
- heroku o now o ngrok *
- servicio de memes !!!!

Promises

```
var p = new Promise((resolve, reject) => { resolve(5) })  
p.then(num => console.log(num)) // 5  
p.then(num => num * 5)  
  .then(num => console.log(num)) // 25
```

- Solo se puede resolver o rechazar una vez
- De ahí en adelante puedes preguntar cuantas veces sea el resultado
- Si una función retorna un valor, el siguiente then obtiene el valor de esta
- para request y otros asyncs...

Slack Apps

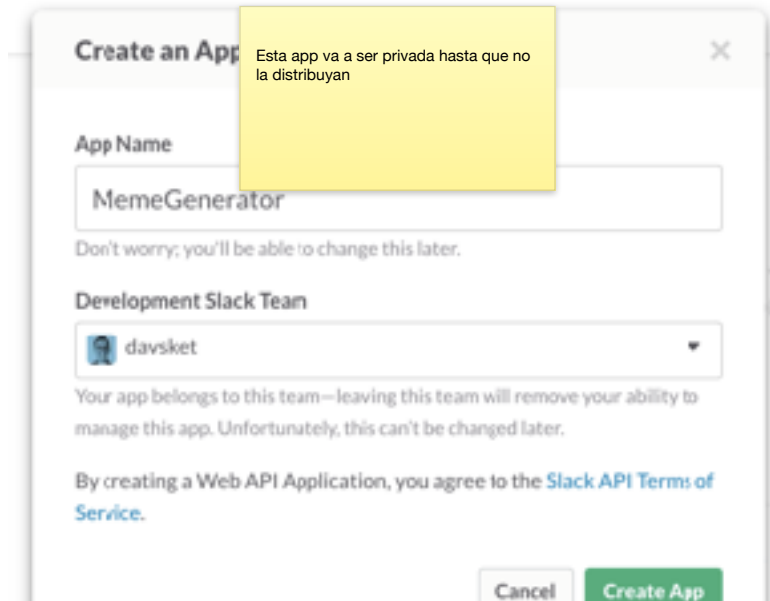
- ya existen muchas apps
- si creas una propia puedes:
 - recibir comandos por medio de texto
 - `/elcomando argumento`
 - `@botname argumentos`
 - mandar respuestas (sync & async)
 - de texto plano
 - con attachments

Slack Bot

- App propia (para escuchar comandos slash con permisos)
- Incoming WebHooks (para mandar mensajes)

Crear App Propia

https://api.slack.com/apps?new_app=1




Create an App ✕

App Name

MemeGenerator

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

Development Slack Team

 davskeet ▼

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

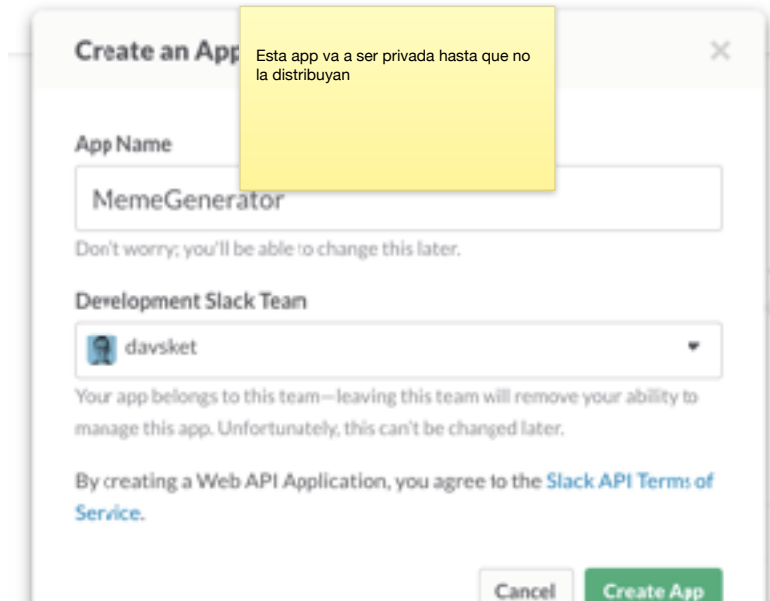
By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel Create App

Esta app va a ser privada hasta que no la distribuyan

Crear App Propia

https://api.slack.com/apps?new_app=1




Create an App ✕

App Name

MemeGenerator

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

Development Slack Team

 davskeet ▼

Your app belongs to this team—leaving this team 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

Esta app va a ser privada hasta que no la distribuyan

App Propia + Slash Cmds

Add features and functionality

Choose and configure the tools you'll need to create your app (or review all [our documentation](#)).

Incoming Webhooks

Post messages from external sources into Slack.

Interactive Messages

Add buttons to your app's messages, and create an interactive experience for users.

Slash Commands

Allow users to perform app actions by typing commands in Slack.

Event Subscriptions

Make it easy for your app to respond to activity in Slack.

Bots

Add a bot to allow users to exchange messages with your app.

Permissions

Configure permissions to allow your app to interact with the Slack API.

App Propia + Slash Cmd

Create New Command

Command

ⓘ

Request URL

ⓘ

Short Description

Usage Hint

Optionally list any parameters that can be passed.

Escape channels, users, and links sent to your app

☒

Unescaped: @user #general

Preview of Autocomplete Entry

Slash Command

```
token=gIkuvNzQIHg97ATvDxagjt0  
team_id=T0001  
team_domain=example  
channel_id=C2147483705  
channel_name=test  
user_id=U2147483697  
user_name=Steve  
command=/weather  
text=94670  
response_url=https://hooks.slack.com/commands/1234/5678
```

App Propia + Add to Team

Basic Information

Building Apps for Slack

Create an app that's just for your team (or build one that can be used by any team) by following the steps below.

Add features and functionality



Install your app to your team

Install your app to your Slack team to test your app and generate the tokens you need to interact with the Slack API. You will be asked to authorize this app after clicking **Install App to Team**.

Install App to Team

Manage Distribution

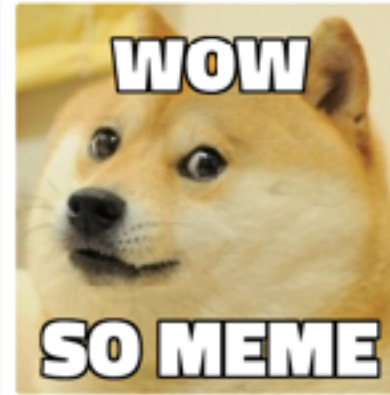


App Propia / Incoming Webhook VS Incoming Webhook App

 MemeGenerator APP 7:33 PM
(301KB)



 davskeet APP 1:21 AM
(282KB)



App Propia + Oauth

Suppar

Settings

Basic Information

Collaborators

Install App

Manage Distribution

Features

Incoming Webhooks

Interactive Messages

Slash Commands

OAuth & Permissions

Event Subscriptions

OAuth & Permissions

OAuth Tokens & Redirect URLs

Tokens for Your Team

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

OAuth Access Token

hexp-3500940031-3500940057-157038291286-d28cfbb9f2b52d76b1f1c0b44a21f2

Copy

Reinstall App

App Propia + Oauth

The screenshot shows the 'OAuth & Permissions' page in the Suppar application. On the left is a sidebar with a 'Suppar' header and a dropdown arrow. Below it are two sections: 'Settings' (with links for Basic Information, Collaborators, Install App, and Manage Distribution) and 'Features' (with links for Incoming Webhooks, Interactive Messages, Slash Commands, OAuth & Permissions, and Event Subscriptions). The 'OAuth & Permissions' link is highlighted in blue. The main content area is titled 'OAuth & Permissions' and contains a section 'OAuth Tokens & Redirect URLs'. Under this is a sub-section 'Tokens for Your Team' with a paragraph explaining that tokens are automatically generated and can be used for authentication, with a 'Learn more' link. Below this is the 'OAuth Access Token' section, which displays a long alphanumeric token in a text box. A red arrow points from the text 'Guardala!!' to the token, and a red oval highlights the token and the 'Copy' button next to it. Below the token is a green 'Reinstall App' button.

Suppar

OAuth & Permissions

Guardala!!

OAuth Tokens & Redirect URLs

Tokens for Your Team

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

OAuth Access Token

heep-3500940031-3500940057-157038291286-d28cfbb9f2b52d76b1f1c0b44a21f2 [Copy](#)

[Reinstall App](#)

App Propia + Permisos

Select Permission Scopes

Add permission by scope or API method...

OTHER

Add commands to davskei.

commands

Post to specific channels in Slack.

incoming-webhook

CHAT

Send messages as MemeGenerator.

chat:write:bot

Send messages as user.

chat:write:user

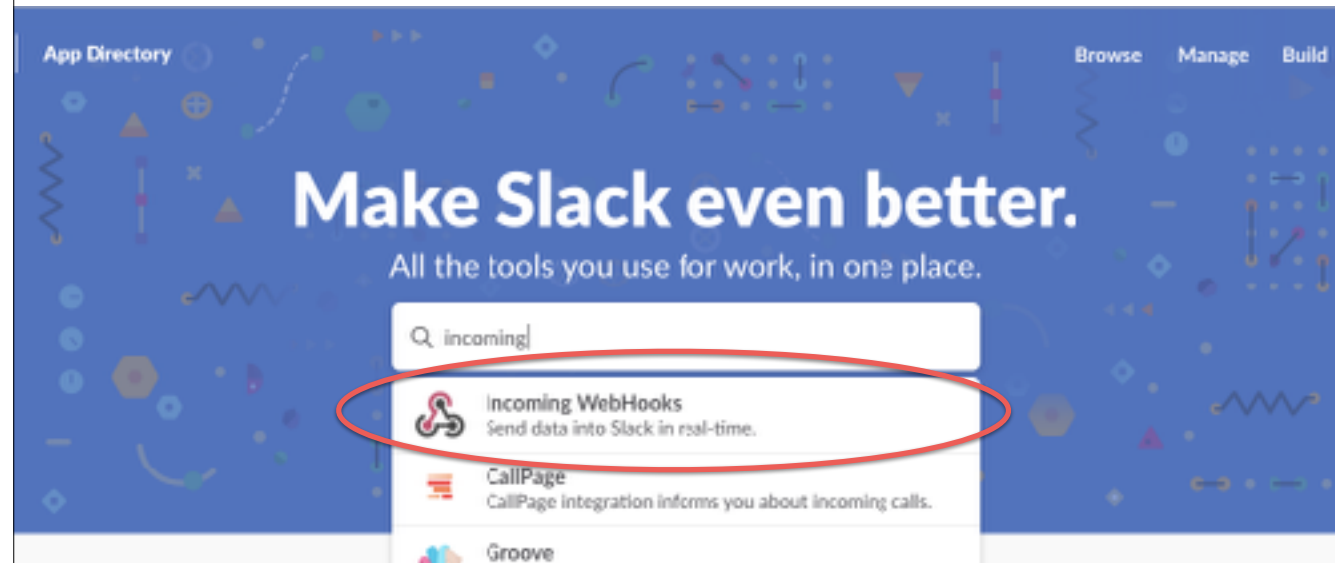
USERS

Access your team's profile information.

users:read


Incoming Webhook

<https://team.slack.com/apps>



Incoming Webhook: add config

[← Browse Apps](#)




Incoming WebHooks

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

[Add Configuration](#)

[Help and support](#)



davsket

Only select team members can install this app.

Incoming Webhook: addconfig

Post to Channel
Start by choosing a channel where your Incoming Webhook will post messages to.

#general ▼

[or create a new channel](#)

Add Incoming WebHooks integration

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).

Incoming Webhook: URL

Integration Settings

Post to Channel
Messages that are sent to the incoming webhook will be posted here.

#general [or create a new channel](#)

Webhook URL
Send your JSON payload to this URL.
[Show setup instructions](#)

[https://hooks.slack.com/services/T03EQTN0X/B4M16JZRU/o9baYhmek6eYw6t-](#) [Copy URL](#) [Regenerate](#)

Descriptive Label
Use this label to provide extra context in your list of integrations (optional).

Optional description of this integration



davsket APP 10:15 AM

(302KB) ▾



davsket APP 1:21 AM

(282KB) ▾



KEYS: 2

Webhook URL

Send your JSON payloads to this URL.

[Show setup instructions](#)

`https://hooks.slack.com/services/T03EQTN0X/B4M16JZRU/o9baYhmek6eYw6t`

[Copy URL](#) • [Regenerate](#)

OAuth Access Token

`xoxp-3500940031-3500943057-157038291284-d28cfbb9f2b52d76b1f1c0b44a21f2` [Copy](#)

CommonJS

hi.js

```
module.exports = 'Hello World'
```

index.js

```
var hi = require('./hi')  
console.log(hi)
```

```
→ commonjs node index.js  
Hello World  
→ commonjs █
```

Variables de Entorno

```
→ commonjs export SLACK_WEBHOOK_URL="https://hooks.slack.com/services/blablabla"  
→ commonjs echo $SLACK_WEBHOOK_URL  
https://hooks.slack.com/services/blablabla  
→ commonjs █
```

```
var url = process.env.SLACK_WEBHOOK_URL  
console.log(url)
```

```
→ commonjs node index.js  
https://hooks.slack.com/services/blablabla  
→ commonjs SLACK_WEBHOOK_URL=yolo.com node index.js  
yolo.com  
→ commonjs █
```

Servicio Memes

<https://memegen.link>

<https://github.com/jacebrowning/memegen>

<https://memegen.link/api>



1. Hacer módulo de Memes

2. Imprimir lista y
meme de la lista

3. Exponer Pagina Web

4. Exponer Pagina
Web pero **Bien**

5. Exponer Pagina Web pero **Cool**

6. Exponer Servicio tipo Bot

Slash Command

POST application/x-www-form-urlencoded

POST Body =

```
token=gJkuvaNzQIHg97ATvDxagjt0  
team_id=T0001  
team_domain=example  
channel_id=C2147483705  
channel_name=test  
user_id=U2147483697  
user_name=Steve  
command=/weather  
text=94&70  
response_url=https://hooks.slack.com/commands/1234/5678
```

Test con ngrok

Hasta el momento...



7. Exponer Servicio tipo Bot **Como Debe Ser**

7. Exponer Servicio tipo Bot

Como Debe Ser

- Obtener info del usuario usando la llave del Oauth
- Enviar meme a Slack usando el Incoming Webhook



Preguntas?

Gracias!

@davsket

[https://github.com/davsket/
slack-bot-experiment](https://github.com/davsket/slack-bot-experiment)