



# Let's build a Slack app!

[github.com/slackapi/WeAreDevs](https://github.com/slackapi/WeAreDevs)



**Requirements:** Python 2/3, pip, ngrok, and a dev Slack workspace



**Who am I?**

Jason Roche



@Roach

**Hi! I'm Jason Roche.** I go by Roche (Roach) because, honestly, there were way too many Jasons born in the 1980s.

I'm a GIF enthusiast and Developer Advocate at  **slack**

My team works on Tools, SDKs, Tutorials, Documentation, Workshops, and other educational content.

# What can the Slack platform do for you?

Slack offers many paths to programmatically interact with your workspaces.

What apps *do* is up to you — perform one distinct task incredibly well or orchestrate concerts between Slack features & your internal accounting service.

Our APIs enable you to:

- Send **rich & timely notifications** with **Webhooks** and the **Web API**.
- Let users **perform workflows** with **interactive messages** & **slash commands**.
- Build **internal integrations** exclusively for your team or distribute with **OAuth**.
- Monitor workspace change in real-time using the **RTM**, and **Events APIs**.

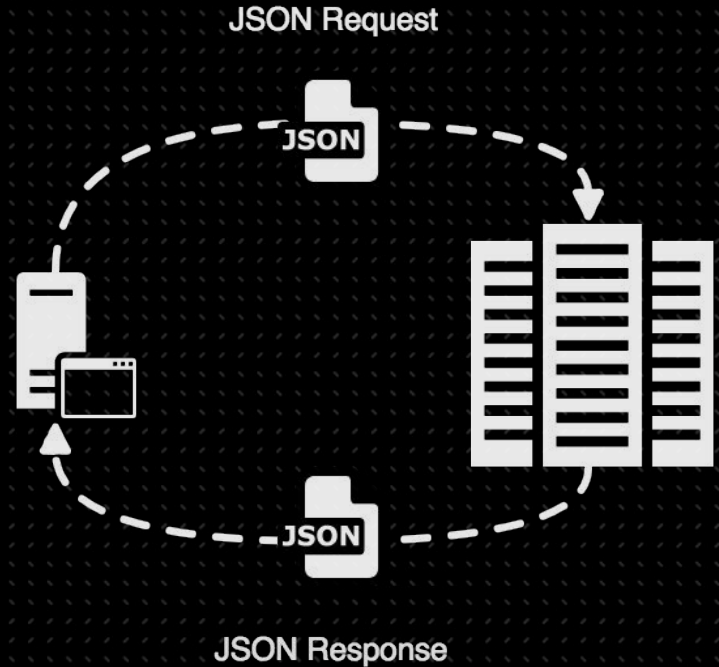
# Slack Web API

Query information, post messages, *change things!* with the Web API.

Use it on the fly for ad-hoc queries or as part of a more complex tapestry of platform features.

It's all HTTP-RPC methods and not quite REST.

[api.slack.com/web](https://api.slack.com/web)  
[api.slack.com/methods](https://api.slack.com/methods)



# Slack Events API

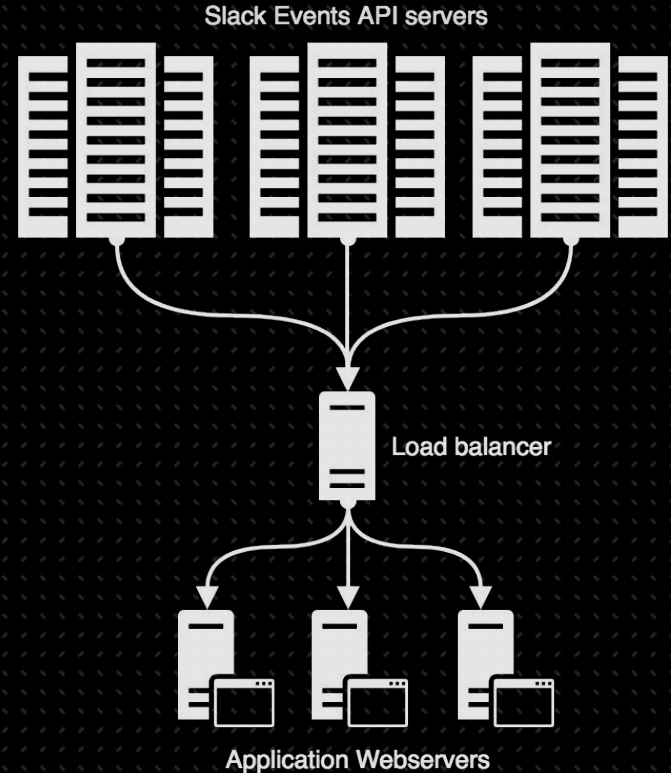
The Events API is a subscription model for apps and bots to process & react to activities in Slack.

All you need is a Slack app and a secure place for Slack to send your events.

## Features:

- Events are sent as HTTP requests
- Standard webserver load-balancing practices
- Only subscribe to the events your app needs
- Limit your app's scopes to only those needed
- Most up to date collection of Slack workspace and user events

[api.slack.com/events-api](https://api.slack.com/events-api)



## Slack RTM API

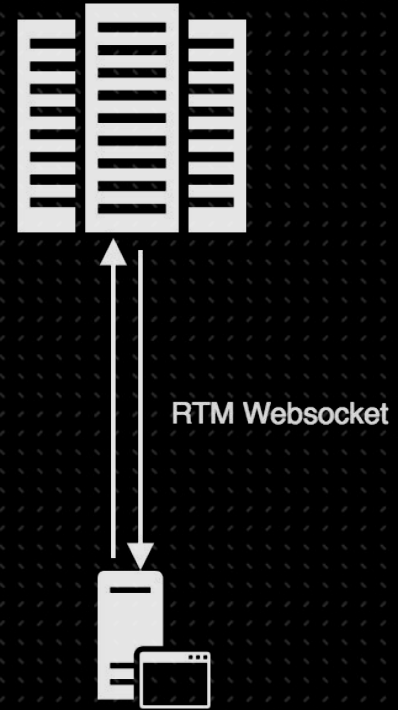
The Real Time Messaging API is a WebSocket-based API that allows you to receive events from Slack in real time and send messages as users. It's sometimes referred to as simply the "RTM API".

Doesn't require an HTTP server

Does require an always-on websocket connection

**May not contain many newer Slack Event types, like bot\_mention**

`api.slack.com/rtm`



Why are we here, together, right now?



We, too, are developers.



# **We are developers and we love Python.**

From one-off scripts to internal tools, we use Python for all sorts of things.

**We know that you probably do, too** and we'd like to make it as seamless as possible for you to build your projects into Slack, or Slack into your projects.

## **Python SlackClient**

The Slack client enables you to interact with the Web API to post messages, reactions, upload files, as well as connect to the RTM API.

## **Slack Events API Adapter**

The Events API Adapter allows you to receive Slack Events API events as subscribable events in your Python app.

## **Python Examples**

We're always adding features to the APIs, and always adding new examples.



**For this project, we only need:**

## **Python SlackClient**

Whether you're building a custom app for your team, or integrating a third party service into your Slack workflows, Slack Developer Kit for Python allows you to leverage the flexibility of Python to get your project up and running as quickly as possible.

Supports: **Web API, RTM API**

## **Python Events API Adapter**

The Slack Events Adapter is a Python-based solution to receive and parse events from Slack's Events API. This library uses an event emitter framework to allow you to easily process Slack events by simply attaching functions to event listeners.

Supports: **Events API**

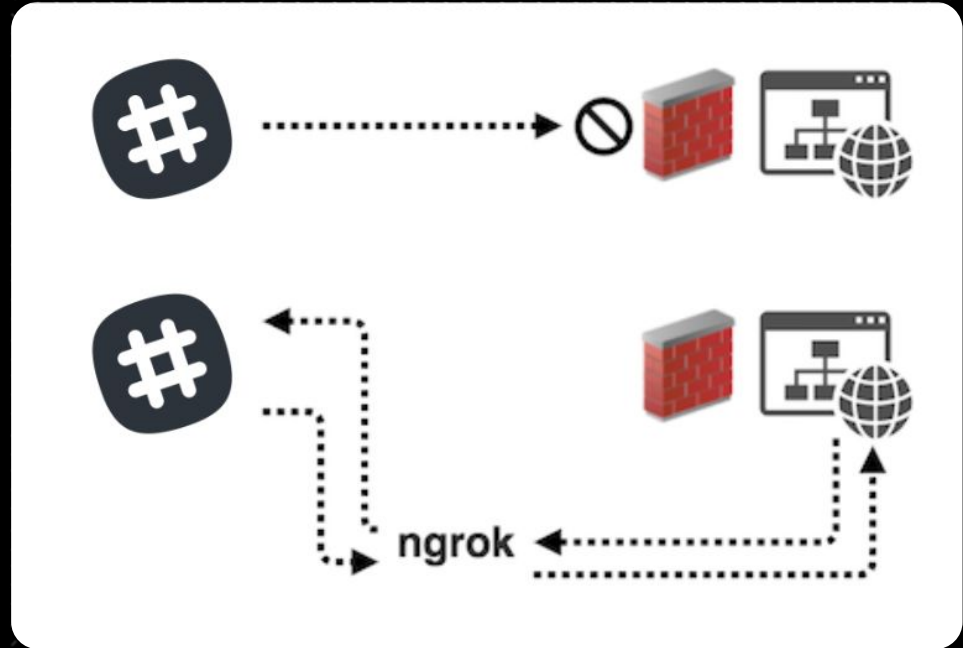
# Ngrok - HTTP tunneling

Ngrok is an awesome tool that allows you to expose a local webserver to the internet without having to expose ports to the world. It's especially helpful for developing webhook-style API apps.

`ngrok.com`

**Note: Run this in a separate terminal window from the python app**

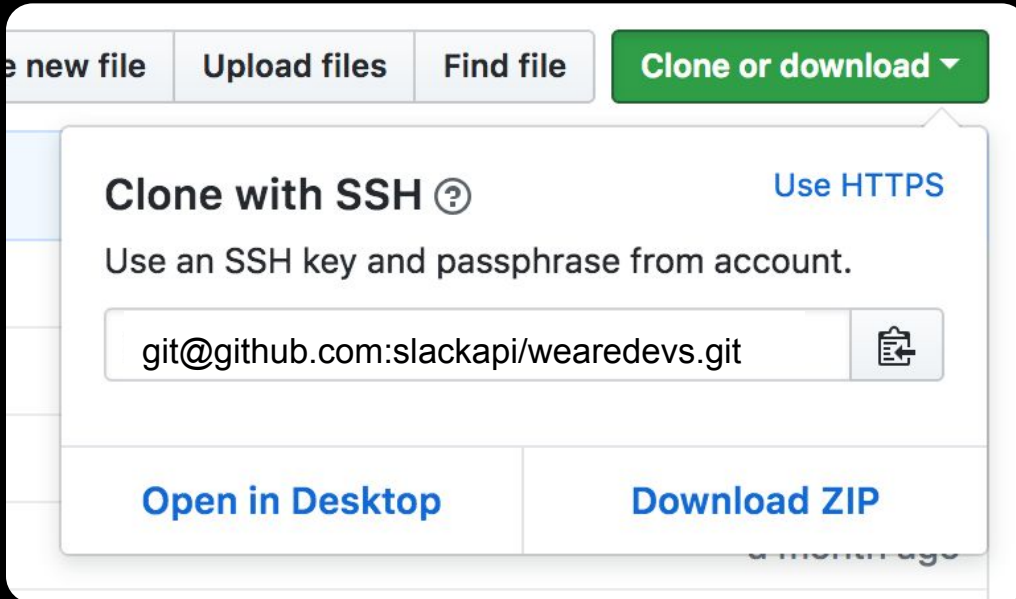
```
$ ngrok http 3000
```



# Download the Pycon example app

Git clone or download the example app repo:

`github.com/slackapi/WeAreDevs`



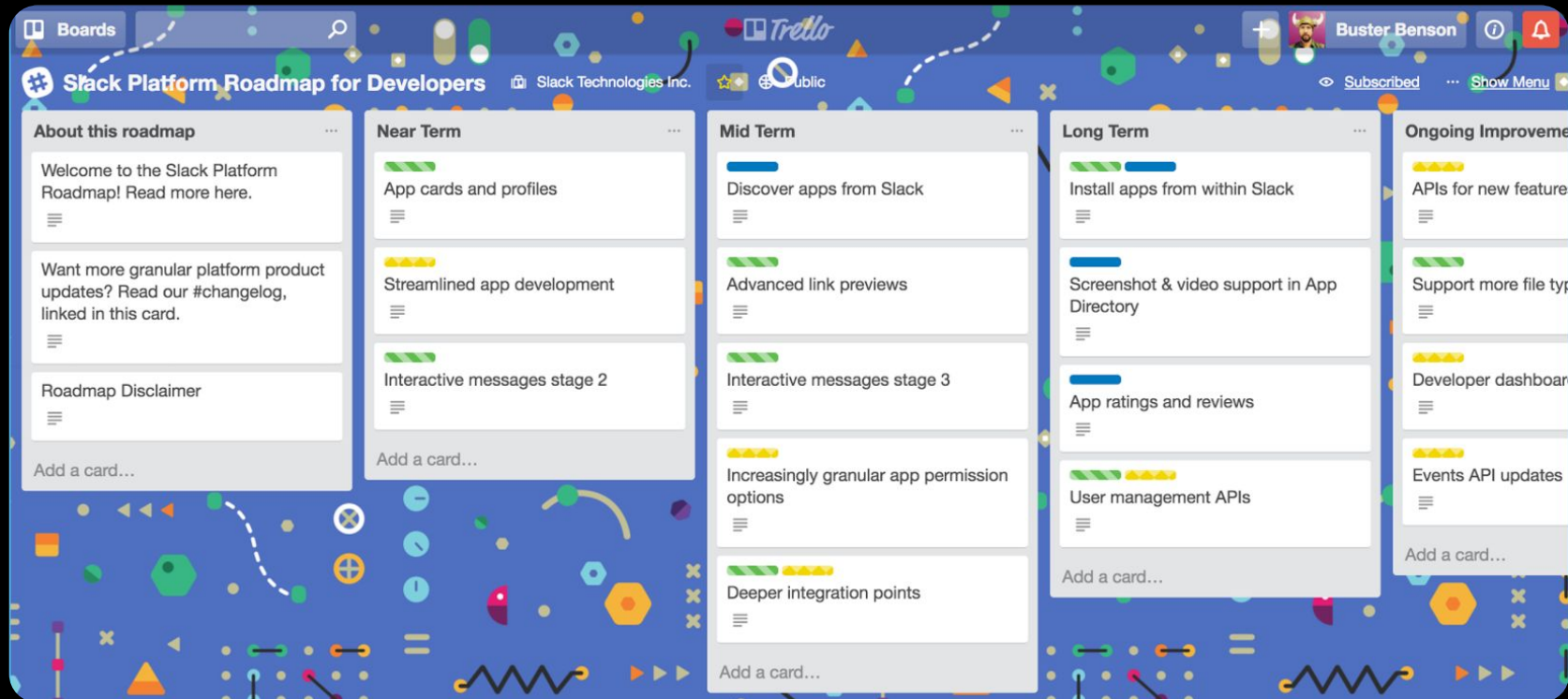
**LIVE DEMO**



**I REGRET NOTHING**

[github.com/slackapi/wearedevs](https://github.com/slackapi/wearedevs)

# Platform roadmap on Trello



## 📌 Stay in touch!



- Read our docs **api.slack.com**
- Check out the **Platform Changelog** for updates

Have a technical question?

Reach out to **developers@slack.com** or chat with us at **@SlackAPI**





I know what you *really* came here for...