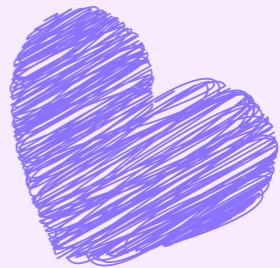


Automating Community

Build-a-Discord Bot



ramonh.dev

badwebsite.club

BAD WEBSITE CLUB!

Learning programming skills can be lonely, and it often feels like there's so much you need to know to call yourself a programmer.

Ramón and Jess have taught over 35,000 learners via our free programming bootcamps* and are really excited to be launching a new social learning project where we'll learn just enough web skills to be dangerous alongside all of you.

*Don't worry, we'll keep doing bootcamps too! Watch this space.

All attendees are expected to abide by our [code of conduct](#)!

Check out our [previous guests](#)!

badwebsite.club

BAD WEBSITE CLUB!

Learning programming skills can be lonely, and it often feels like there's so much you need to know to call yourself a programmer.

Ramón and Jess have taught over 35,000 learners via our free programming bootcamps* and are really excited to be launching a new social learning project where we'll learn just enough web skills to be dangerous alongside all of you.

Wait wait wait

• [Twitch!](#)

• [Youtube](#)

For precise dates, [check out our schedule!](#)

Check out the source code for this website, or propose changes to it, on [GitHub](#).

► [How much will this cost?](#)

► [Really free? How can it be free?](#)

► [Will you help me find a job in tech?](#)

► [What if the scheduled times don't work for me?](#)

► [This website sucks?](#)

The art on our YouTube channel and blog posts are made by the fabulous [Kiri \(Instagram\)](#)



I'm Ramón. (he/him)

From , living in 

Co-Founder: [BadWebsite.Club](#)

DevRel Strategy Consultant

egghead Instructor

Community member

Mozilla tech speaker alum

Kids' coding coach

Coding live streamer

badwebsite.club







Ramón (he/him) 03/04/2023 17:45

Hello! Welcome to the Bad Website Club Discord server ✨


You'll find all you need to collab with one another and ask questions!


A few points before getting started:














- Hop into the [#introductions](#) channel and introduce yourself!
- All recordings live on our YouTube Channel: <https://www.youtube.com/@BadWebsiteClub>
- We expect everyone to adhere to our code of conduct. Please read it here: <https://bit.ly/bwc-coc>
- Please do not send spam
- Any issues found here that break out code of conduct, please let the [@mods](#) or [@admin](#) know.








New Members

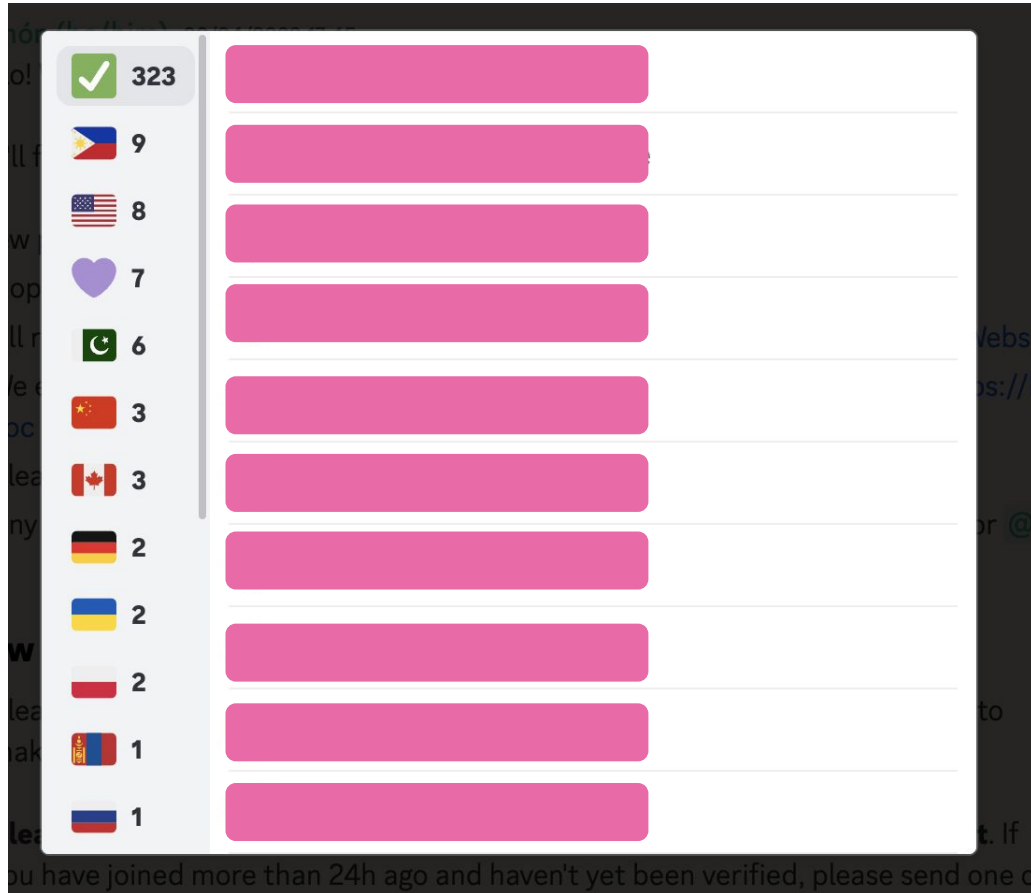
Please note that joining this server will assign you an [@unverified](#) role. This is to make sure we're securing this server.

Please react to this message with  and an [@admin](#) will verify you via bot. If you have joined more than 24h ago and haven't yet been verified, please send one of us a message!

Be sure to share and be kind with one another.  We're all in this together! (edited)

 323
  1
  3
  1
  7
  1
  1
  3
  9
  1
  8
  1
  6

 2
  2
  2
  1
  1
  1
  1



you have joined more than 24h ago and haven't yet been verified, please send one o

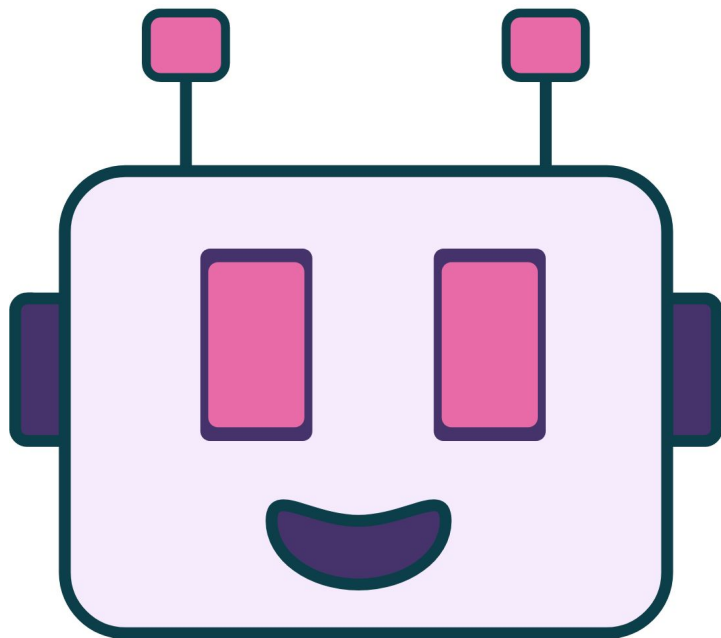
**On two separate
windows/tabs, btw**

It was cumbersome

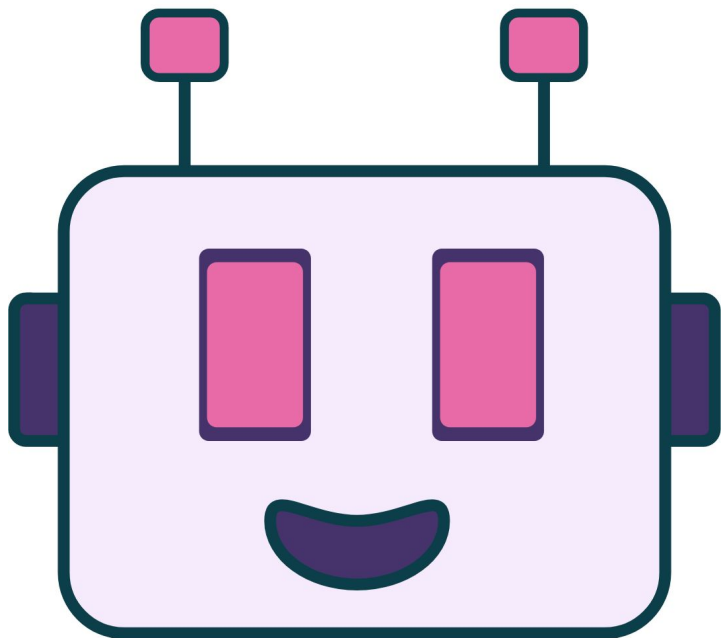


I started making mistakes

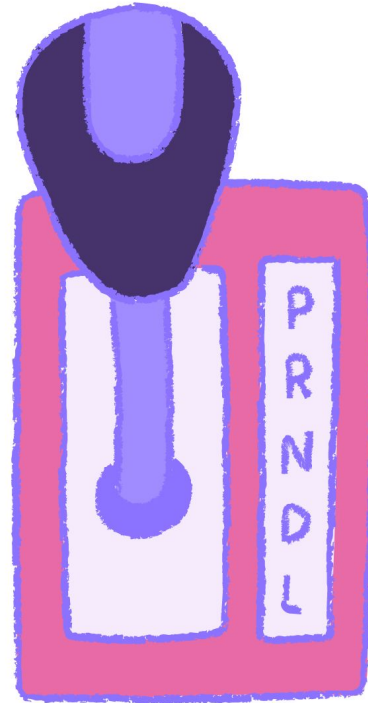
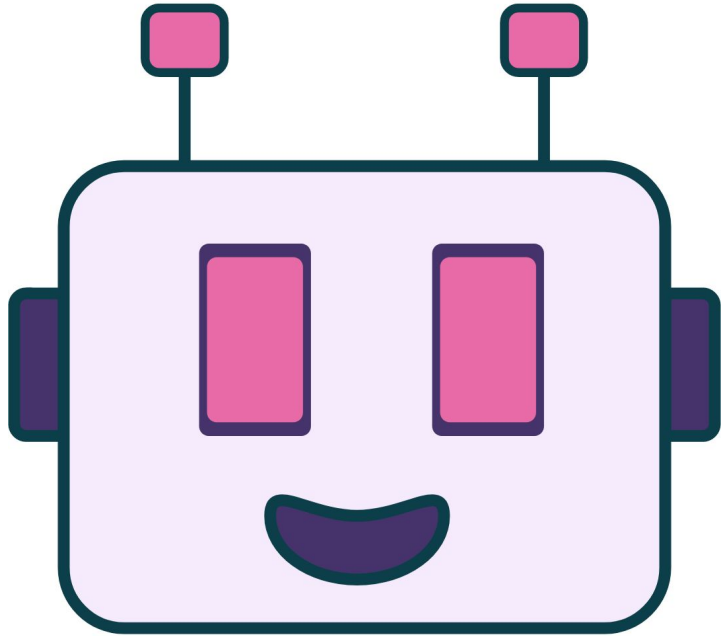
We're making a bot!



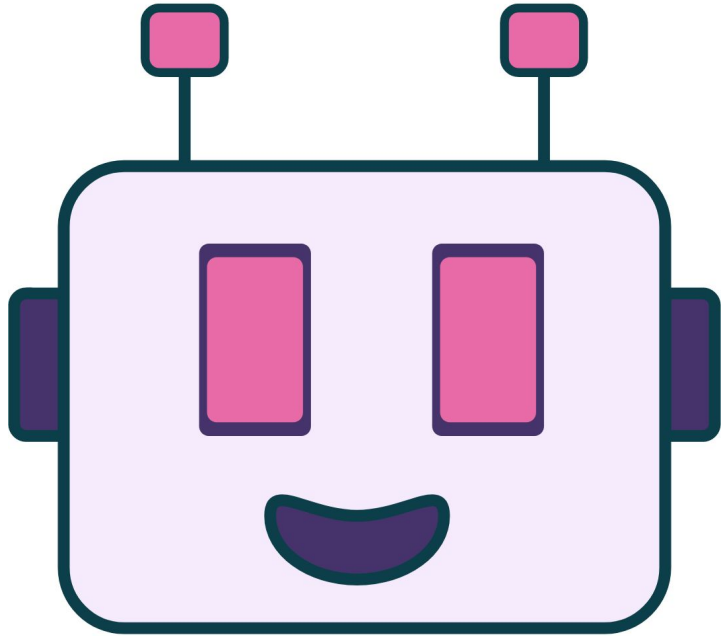
What is a Discord bot?



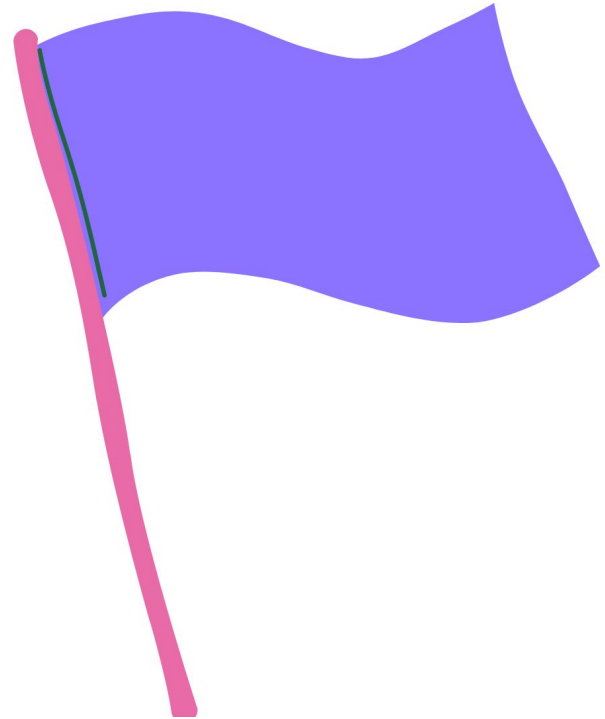
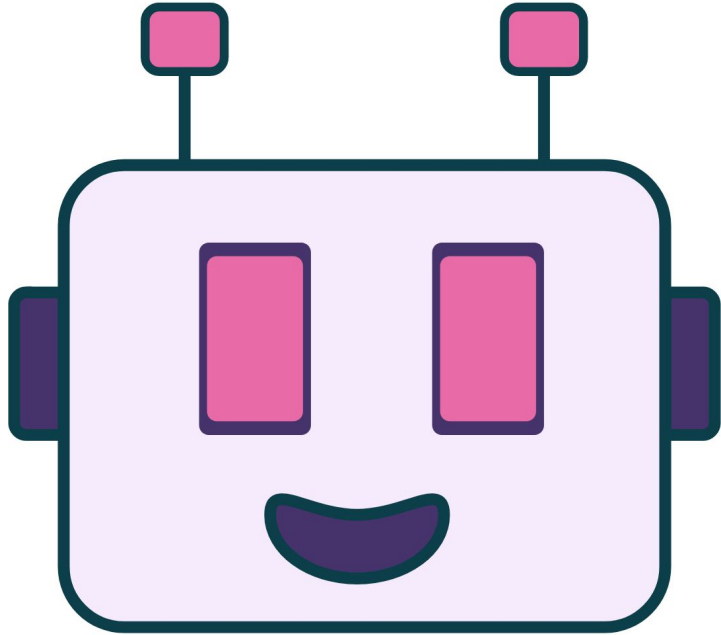
Automation!



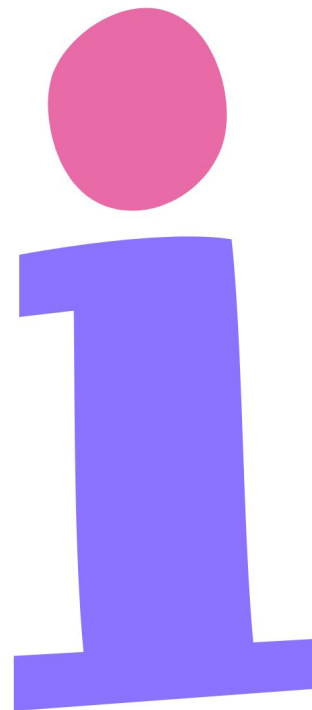
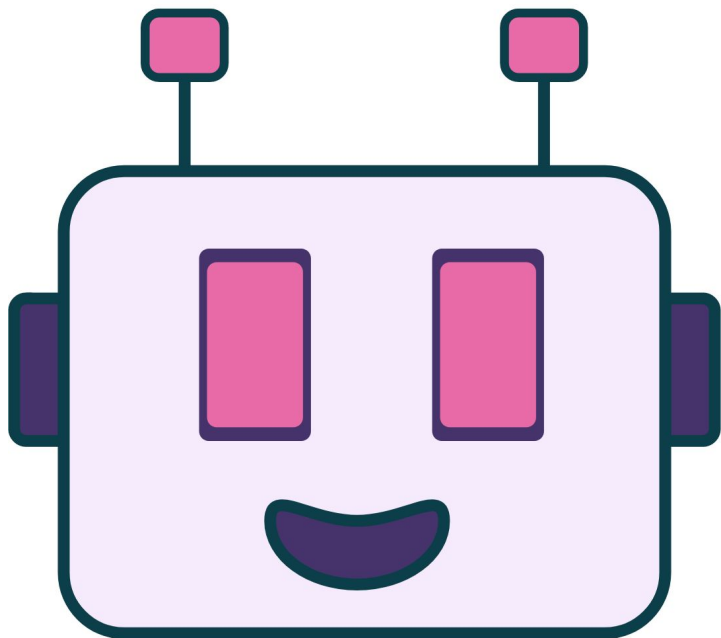
Interaction!



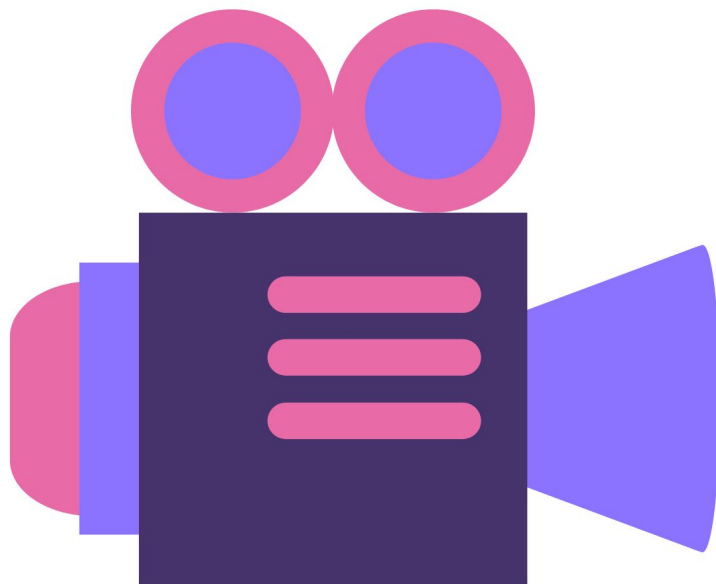
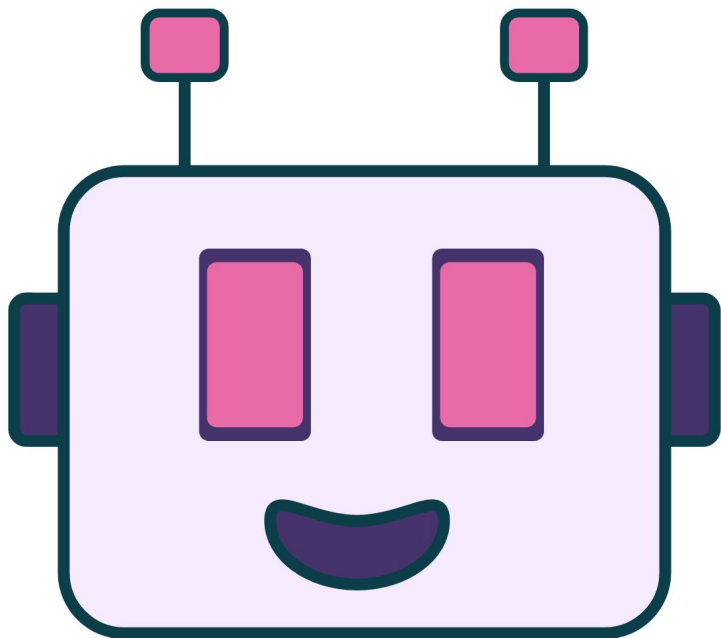
Moderation!



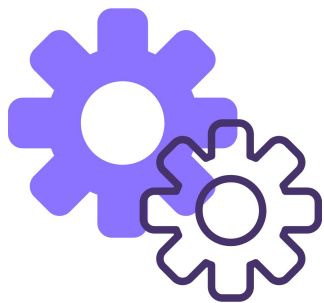
Information!



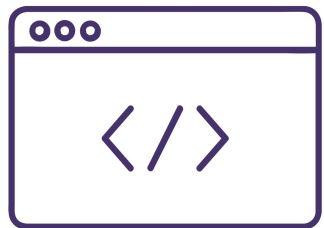
Entertainment!



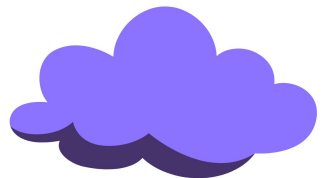
Cool,
so how do we make bots?



Registered Bot



Bot Codebase



Cloud Deployment

discord.com/developers

DEVELOPER PORTAL

Applications

Teams

Server Insights

Embed Debugger

Documentation

Applications

New Application



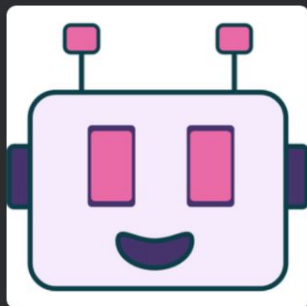
Find the perfect feature set for your game in our Game SDK, and sign up for Server Commerce to start selling directly in your server. Get started by creating a new application. We can't wait to see what you make!

Sort By: Date Created

Small

Large

My Applications



verify-bot

discord.js.org

You are reading the documentation for the **next** version of discord.js. Documentation for v13/v14+ has been moved to [old.discordjs.dev](#)

The **most popular** way to build Discord bots.

discord.js is a powerful Node.js module that allows you to interact with the Discord API very easily. It takes a much more object-oriented approach than most other JS Discord libraries, making your bot's code significantly tidier and easier to comprehend.

[Docs](#)

[Module docs](#)

[Guide](#) 

[GitHub](#) 

```
> npm install discord.js 
```



Powered by Vercel



Powered by Workers

fly.io



[Articles](#) [Docs](#) [Community](#) [Status](#) [Pricing](#)

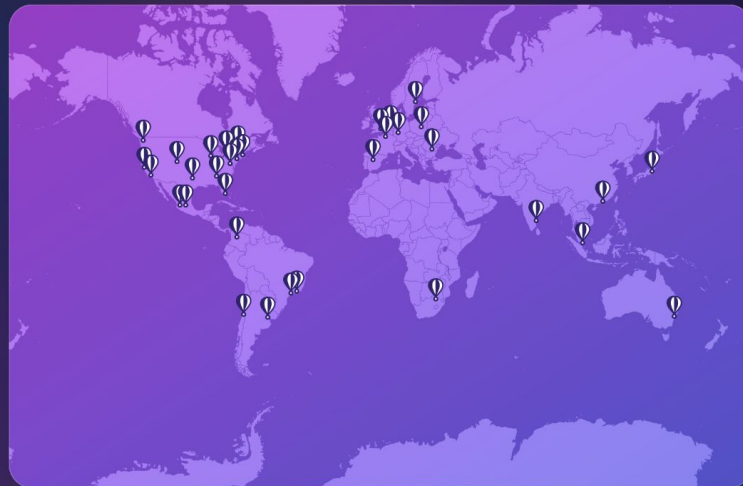
[Sign In](#)

[Sign Up](#)

Launch Apps Near Users

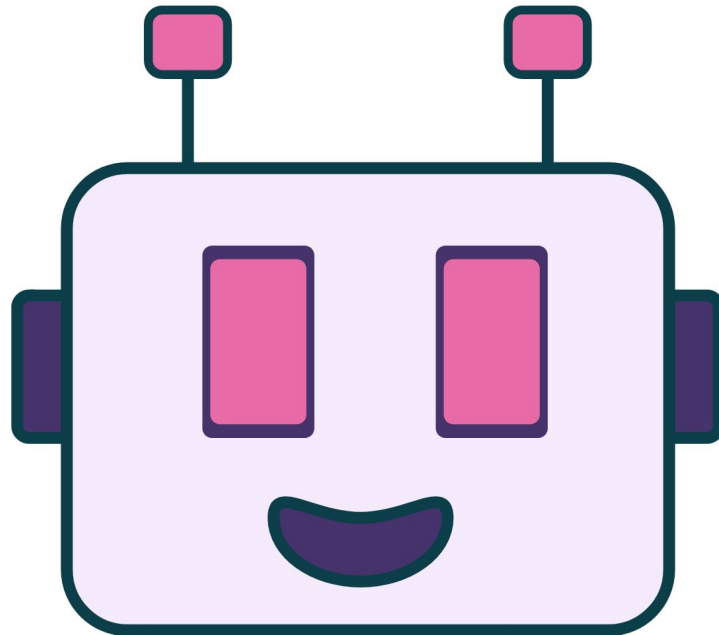
Fly.io transforms containers into micro-VMs that run on our hardware in 30+ regions on six continents.*

[Launch an App for Free >](#)



* Antarctica coming soon.

verify-bot



Part 1

The Discord Dev. Portal

You're a Developer Now

Server Boost

Subscriptions

Gift Inventory

Billing

APP SETTINGS

Appearance NEW

Accessibility

Voice & Video

Text & Images

Notifications

Keybinds

Language

Streamer Mode

Advanced

Advanced

Developer Mode

Developer Mode exposes context menu items helpful for people writing bots using the [Discord API](#).

Hardware Acceleration

Turns on Hardware Acceleration, which uses your GPU to make Discord smoother. Turn this off if you are experiencing frame drops.

Application Test Mode

Enter your Discord application ID to enable test mode for that application.

Developer Activity Shelf

When enabled, the activity shelf will show all activities your user and user's teams have access to.

discord.com/developers

DEVELOPER PORTAL

Applications

Teams

Server Insights

Embed Debugger

Documentation

Applications

New Application 

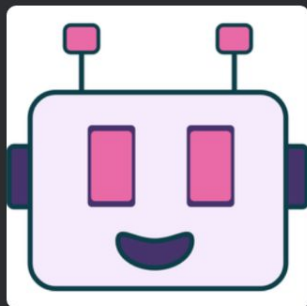
Find the perfect feature set for your game in our Game SDK, and sign up for Server Commerce to start selling directly in your server. Get started by creating a new application. We can't wait to see what you make!

Sort By: Date Created ▾

⋮ Small

⋮ Large

My Applications



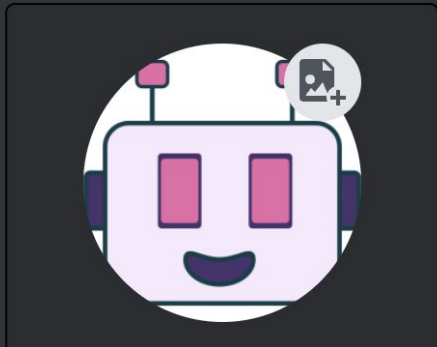
verify-bot

Grab your keys

Build-A-Bot

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to destroy).

ICON



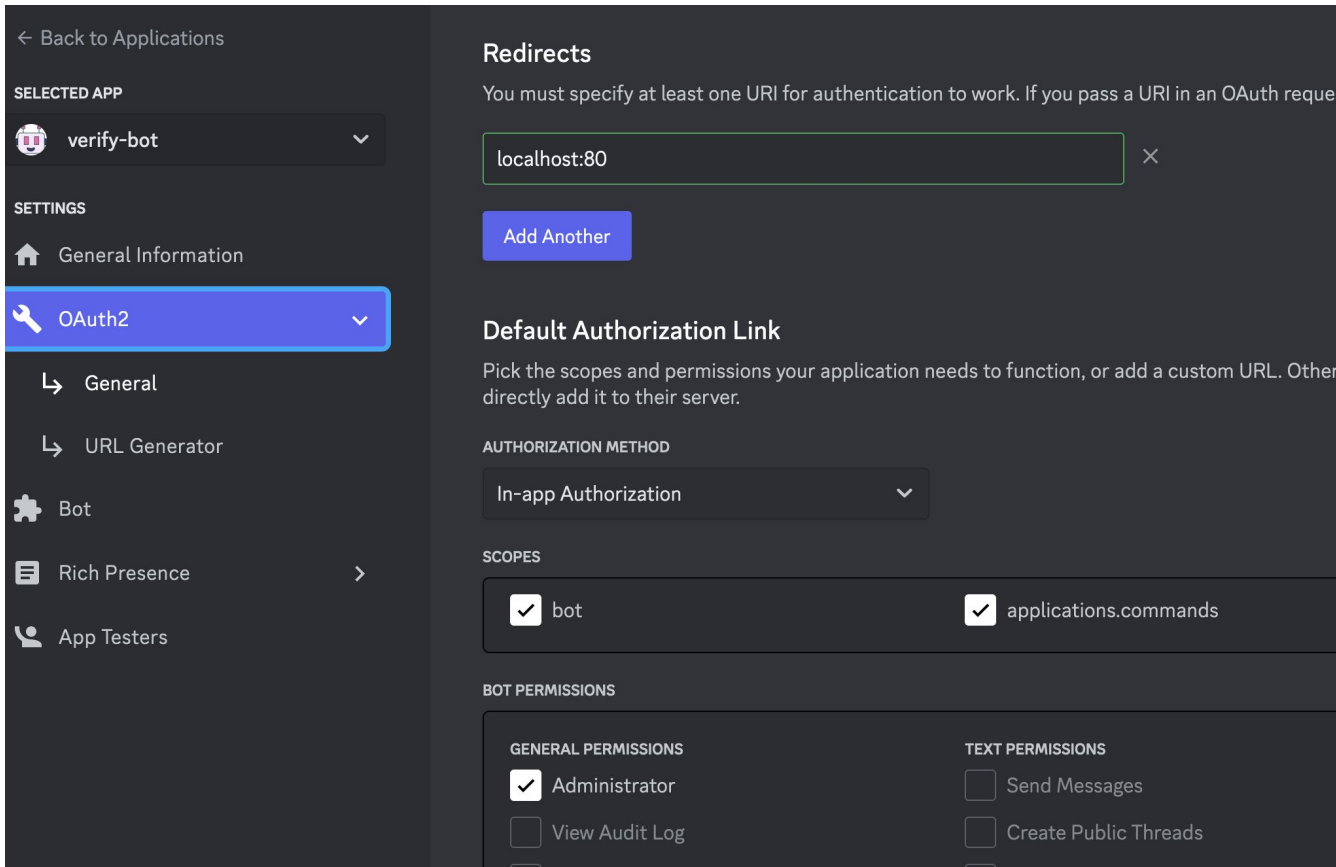
USERNAME

TOKEN

For security purposes, tokens can only be viewed once, when created. If you forgot or lost access to your token, please regenerate a new one.


[Reset Token](#)

Give it permissions



← Back to Applications

SELECTED APP

 verify-bot

SETTINGS

- General Information
- OAuth2**
- General
- URL Generator
- Bot
- Rich Presence
- App Testers

Redirects

You must specify at least one URI for authentication to work. If you pass a URI in an OAuth request, you must specify a redirect URI.

localhost:80

Add Another

Default Authorization Link

Pick the scopes and permissions your application needs to function, or add a custom URL. Other users can also be added to their server.

AUTHORIZATION METHOD

In-app Authorization

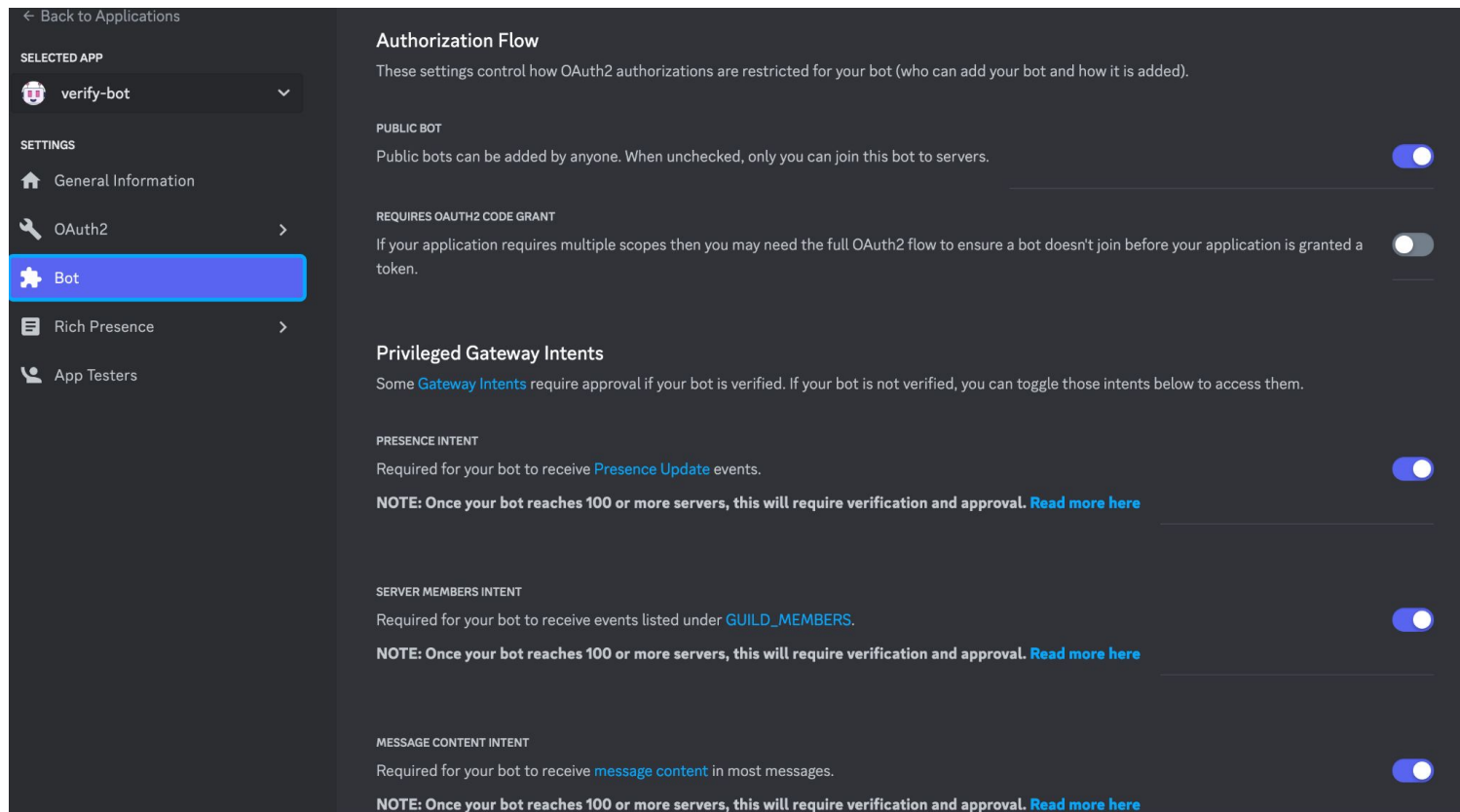
SCOPES

- bot
- applications.commands

BOT PERMISSIONS


GENERAL PERMISSIONS	TEXT PERMISSIONS
<input checked="" type="checkbox"/> Administrator	<input type="checkbox"/> Send Messages
<input type="checkbox"/> View Audit Log	<input type="checkbox"/> Create Public Threads

Give it permissions



← Back to Applications

SELECTED APP

 verify-bot

SETTINGS

- General Information
- OAuth2
- Bot**
- Rich Presence
- App Testers

Authorization Flow

These settings control how OAuth2 authorizations are restricted for your bot (who can add your bot and how it is added).

PUBLIC BOT

Public bots can be added by anyone. When unchecked, only you can join this bot to servers.

REQUIRES OAUTH2 CODE GRANT

If your application requires multiple scopes then you may need the full OAuth2 flow to ensure a bot doesn't join before your application is granted a token.

Privileged Gateway Intents

Some [Gateway Intents](#) require approval if your bot is verified. If your bot is not verified, you can toggle those intents below to access them.

PRESENCE INTENT

Required for your bot to receive [Presence Update](#) events.

NOTE: Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)

SERVER MEMBERS INTENT

Required for your bot to receive events listed under [GUILD_MEMBERS](#).

NOTE: Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)

MESSAGE CONTENT INTENT

Required for your bot to receive [message content](#) in most messages.

NOTE: Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)

Part 2

The Codebase!

```

  ✓ VERIFY-BOT
    ✓ commands
      JS verify.js
      > node_modules
      ⚙ .gitignore
      JS index.js
      {} package-lock.json
      {} package.json
      ⓘ README.md

```


index.js

```
// Require the necessary discord.js classes
const fs = require("node:fs");
const path = require("node:path");
const { Client, Collection, Events, GatewayIntentBits } = require("discord.js");
const token = process.env.DISCORD_TOKEN;

// Create a new client instance
const client = new Client({ intents: [GatewayIntentBits.Guilds] });
client.commands = new Collection();
const commandsPath = path.join(__dirname, "commands");
const commandFiles = fs
  .readdirSync(commandsPath)
  .filter((file) => file.endsWith(".js"));
```

index.js

```
for (const file of commandFiles) {
  const filePath = path.join(commandsPath, file);
  const command = require(filePath);

  // Set a new item in the Collection with the key as the command
  // name and the value as the exported module
  if ("data" in command && "execute" in command) {
    client.commands.set(command.data.name, command);
  } else {
    console.log(
      `[WARNING] The command at ${filePath}
      is missing a required "data" or "execute" property.`
    );
  }
}
```

index.js

```
client.on(Events.InteractionCreate, async (interaction) => {
  if (!interaction.isChatInputCommand()) return;

  const command = interaction.client.commands.get(interaction.commandName);

  if (!command) {
    console.error(`No command matching ${interaction.commandName} was found.`);
    return;
  }

  try {
    await command.execute(interaction);
  } catch (error) {
    console.error(error);
    if (interaction.replied || interaction.deferred) {
      await interaction.followUp({
        content: "There was an error while executing this command!",
        ephemeral: true,
      });
    } else {
      await interaction.reply({
        content: "There was an error while executing this command!",
        ephemeral: true,
      });
    }
  }
});
```

index.js

```
// When the client is ready, run this code (only once)
// We use 'c' for the event parameter to keep it separate
// from the already defined 'client'
client.once(Events.ClientReady, (c) => {
  console.log(`Ready! Logged in as ${c.user.tag}`);
});

// Log in to Discord with your client's token
client.login(token);
```

verify.js

```
const { SlashCommandBuilder } = require("discord.js");
const channelID = process.env.CHANNEL_ID;
const messageID = process.env.MESSAGE_ID;
const emojiID = process.env.EMOJI_ID;
const serverID = process.env.SERVER_ID;

module.exports = {
  data: new SlashCommandBuilder()
    .setName("verify")
    .setDescription("Replies with whether users are verified!"),
  async execute(client, interaction) {
    const channel = await client.channels.fetch(channelID);
    const message = await channel.messages.fetch(messageID);
    const guild = await client.guilds.fetch(serverID);
    const reaction = message.reactions.cache.get(emojiID);
    const users = await reaction.users.fetch();
```

verify.js

```
const { SlashCommandBuilder } = require("discord.js");
const channelID = process.env.CHANNEL_ID;
const messageID = process.env.MESSAGE_ID;
const emojiID = process.env.EMOJI_ID;
const serverID = process.env.SERVER_ID;

module.exports = {
  data: new SlashCommandBuilder()
    .setName("verify")
    .setDescription("Replies with whether users are verified!"),
  async execute(client, interaction) {
    const channel = await client.channels.fetch(channelID);
    const message = await channel.messages.fetch(messageID);
    const guild = await client.guilds.fetch(serverID);
    const reaction = message.reactions.cache.get(emojiID);
    const users = await reaction.users.fetch();
```

verify.js

```
users.forEach(async (user) => {
  try {
    // get the member object as users don't have roles
    const member = await guild.members.fetch(user.id);
    if (member) {
      // See if they're unverified
      const role = member.roles.cache.find((r) => r.name === "unverified");
      if (role) {
        console.log(user.displayName, "verified");

        member.roles.remove(role.id);
      }
    } else {
      console.log(member, user.id, "not found, not verifying?");
    }
  } catch (e) {
    console.log("err: unknown member", user.displayName);
    await reaction?.users.remove(user.id);
  }
});
await interaction.reply(`Verifying members`);
},
};
```

verify.js

```
users.forEach(async (user) => {
  try {
    // get the member object as users don't have roles
    const member = await guild.members.fetch(user.id);
    if (member) {
      // See if they're unverified
      const role = member.roles.cache.find((r) => r.name === "unverified");
      if (role) {
        console.log(user.displayName, "verified");

        member.roles.remove(role.id);
      }
    } else {
      console.log(member, user.id, "not found, not verifying?");
    }
  } catch (e) {
    console.log("err: unknown member", user.displayName);
    await reaction?.users.remove(user.id);
  }
});
await interaction.reply(`Verifying members`);
},
};
```


verify.js

```
users.forEach(async (user) => {
  try {
    // get the member object as users don't have roles
    const member = await guild.members.fetch(user.id);
    if (member) {
      // See if they're unverified
      const role = member.roles.cache.find((r) => r.name === "unverified");
      if (role) {
        console.log(user.displayName, "verified");

        member.roles.remove(role.id);
      }
    } else {
      console.log(member, user.id, "not found, not verifying?");
    }
  } catch (e) {
    console.log("err: unknown member", user.displayName);
    await reaction?.users.remove(user.id);
  }
});
await interaction.reply(`Verifying members`);
};
```

Part 3

Deployment!

fly.io



[Articles](#) [Docs](#) [Community](#) [Status](#) [Pricing](#)

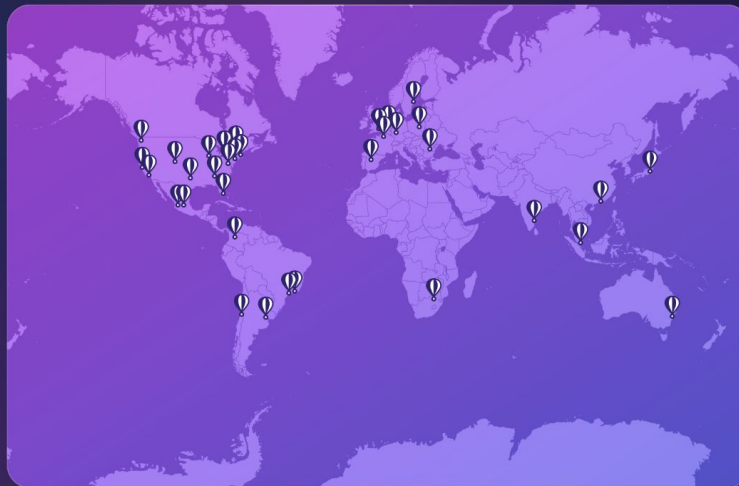
[Sign In](#)

[Sign Up](#)

Launch Apps Near Users

Fly.io transforms containers into micro-VMs that run on our hardware in 30+ regions on six continents.*

[Launch an App for Free >](#)



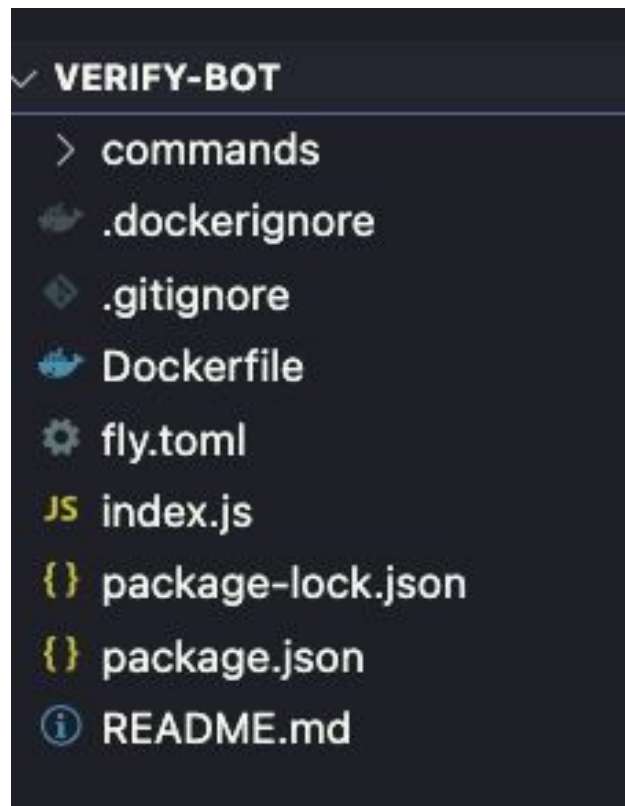
* Antarctica coming soon.

1. Install the flyctl tool

**2. flyctl auth login
or
flyctl auth signup**

3. flyctl launch

We got new files!



3. fly secrets set
DISCORD_TOKEN=super_secret

3. flyctl deploy

fly.io Dashboard

Dashboard Resources Account

personal > bwc-verify-bot > Overview
Deployed

Overview

Machines

Volumes

Activity

Metrics

Monitoring

Sentry Errors

Certificates

Secrets

Tokens

Settings

Average Data In (1h)
150 B/s

Average Data Out (1h)
27 B/s

Average Memory Used...
96 MB/228 MB

Average Load (1h)
0.00

Application Information

Personal details and application.

Hostname	App Type
https://bwc-verify-bot.fly.dev	Apps v2

Process groups

app

Machine Sizes

shared-1x-cpu@256MB

IP addresses

66.241.124.33

i
Shared v4

Activity

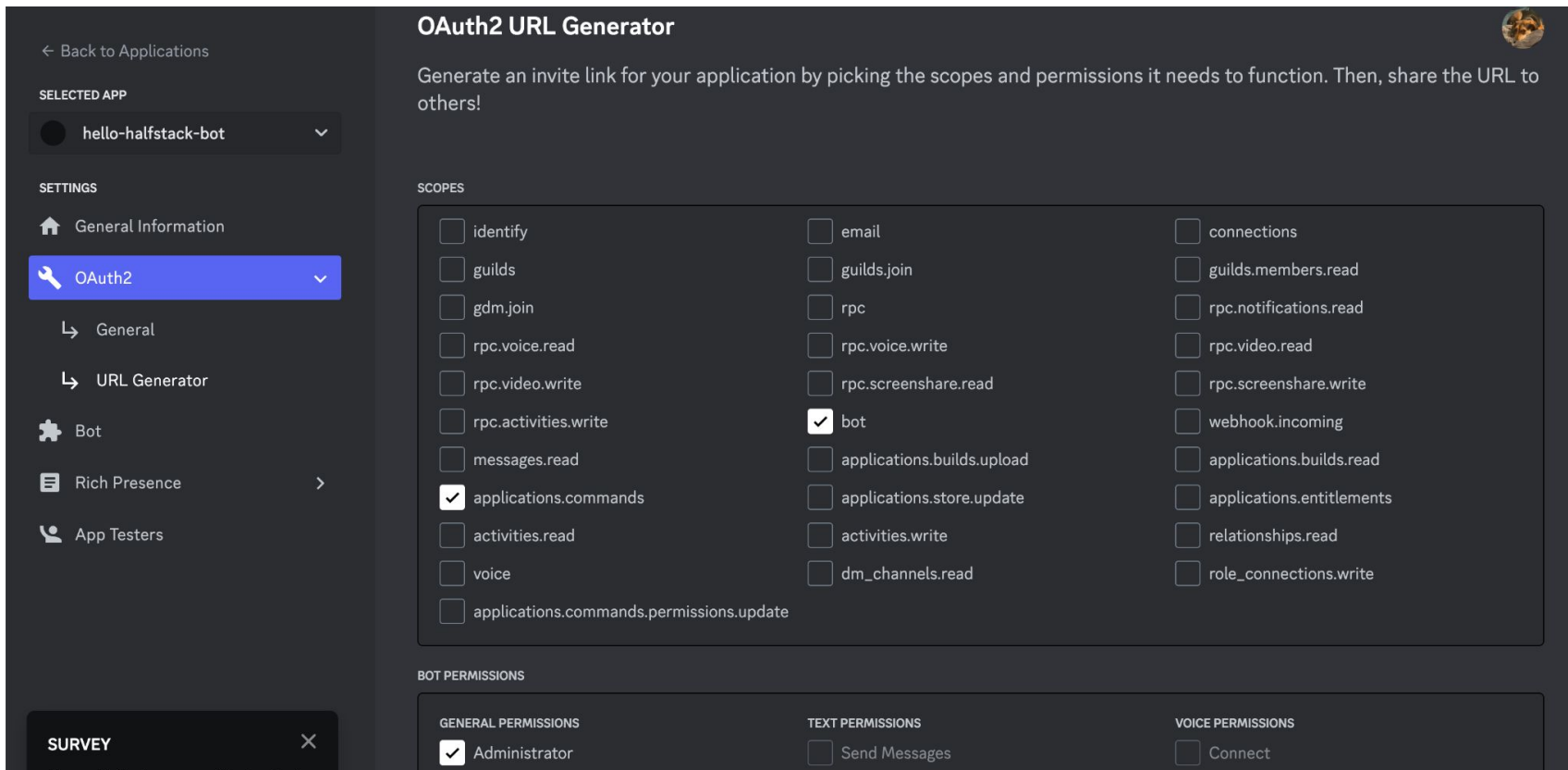
Release
by Ramón Huidobro 20 days
Released v79

Release
by Ramón Huidobro 20 days
Released v78

Release
by Ramón Huidobro 20 days
Released v77

Release
by Ramón Huidobro 21 days
Released v76

Get URL and add it to Discord



The screenshot shows the 'OAuth2 URL Generator' page for the application 'hello-halfstack-bot'. The interface is dark-themed and includes a left sidebar with navigation options like 'General Information', 'OAuth2', 'General', 'URL Generator', 'Bot', 'Rich Presence', and 'App Testers'. The main content area is titled 'OAuth2 URL Generator' and contains instructions: 'Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!'. Below the instructions, there are two sections: 'SCOPES' and 'BOT PERMISSIONS'. The 'SCOPES' section contains a grid of checkboxes for various permissions, with 'bot' and 'applications.commands' checked. The 'BOT PERMISSIONS' section has three sub-sections: 'GENERAL PERMISSIONS' (with 'Administrator' checked), 'TEXT PERMISSIONS' (with 'Send Messages' unchecked), and 'VOICE PERMISSIONS' (with 'Connect' unchecked). A 'SURVEY' button is visible in the bottom left corner.

← Back to Applications

SELECTED APP

hello-halfstack-bot

SETTINGS

General Information

OAuth2

General

URL Generator

Bot

Rich Presence

App Testers

OAuth2 URL Generator

Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!

SCOPES

<input type="checkbox"/> identify	<input type="checkbox"/> email	<input type="checkbox"/> connections
<input type="checkbox"/> guilds	<input type="checkbox"/> guilds.join	<input type="checkbox"/> guilds.members.read
<input type="checkbox"/> gdm.join	<input type="checkbox"/> rpc	<input type="checkbox"/> rpc.notifications.read
<input type="checkbox"/> rpc.voice.read	<input type="checkbox"/> rpc.voice.write	<input type="checkbox"/> rpc.video.read
<input type="checkbox"/> rpc.video.write	<input type="checkbox"/> rpc.screenshare.read	<input type="checkbox"/> rpc.screenshare.write
<input type="checkbox"/> rpc.activities.write	<input checked="" type="checkbox"/> bot	<input type="checkbox"/> webhook.incoming
<input type="checkbox"/> messages.read	<input type="checkbox"/> applications.builds.upload	<input type="checkbox"/> applications.builds.read
<input checked="" type="checkbox"/> applications.commands	<input type="checkbox"/> applications.store.update	<input type="checkbox"/> applications.entitlements
<input type="checkbox"/> activities.read	<input type="checkbox"/> activities.write	<input type="checkbox"/> relationships.read
<input type="checkbox"/> voice	<input type="checkbox"/> dm_channels.read	<input type="checkbox"/> role_connections.write
<input type="checkbox"/> applications.commands.permissions.update		

BOT PERMISSIONS

GENERAL PERMISSIONS	TEXT PERMISSIONS	VOICE PERMISSIONS
<input checked="" type="checkbox"/> Administrator	<input type="checkbox"/> Send Messages	<input type="checkbox"/> Connect

SURVEY ✕

Back in Discord...



 Ramón (he/him) used `/verify`

verify-bot BOT Today at 00:22

Verifying members

fly.io Monitoring

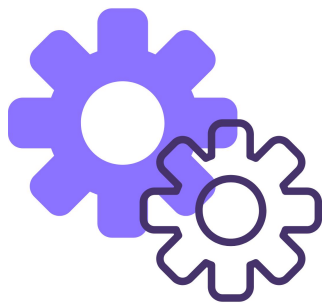
personal > bwc-verify-bot > Monitoring Deployed

- Overview
- Machines
- Volumes
- Activity
- Metrics
- Monitoring
- Sentry Errors
- Certificates
- Secrets
- Tokens
- Settings

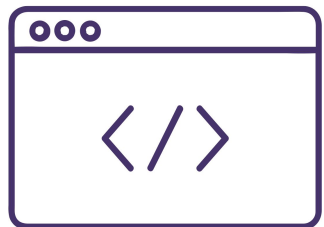
Monitoring All regions All instances

```
$ fly logs -a bwc-verify-bot  
Waiting for logs...
```

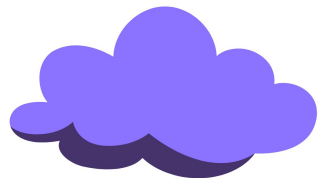
Machines



Registered Bot



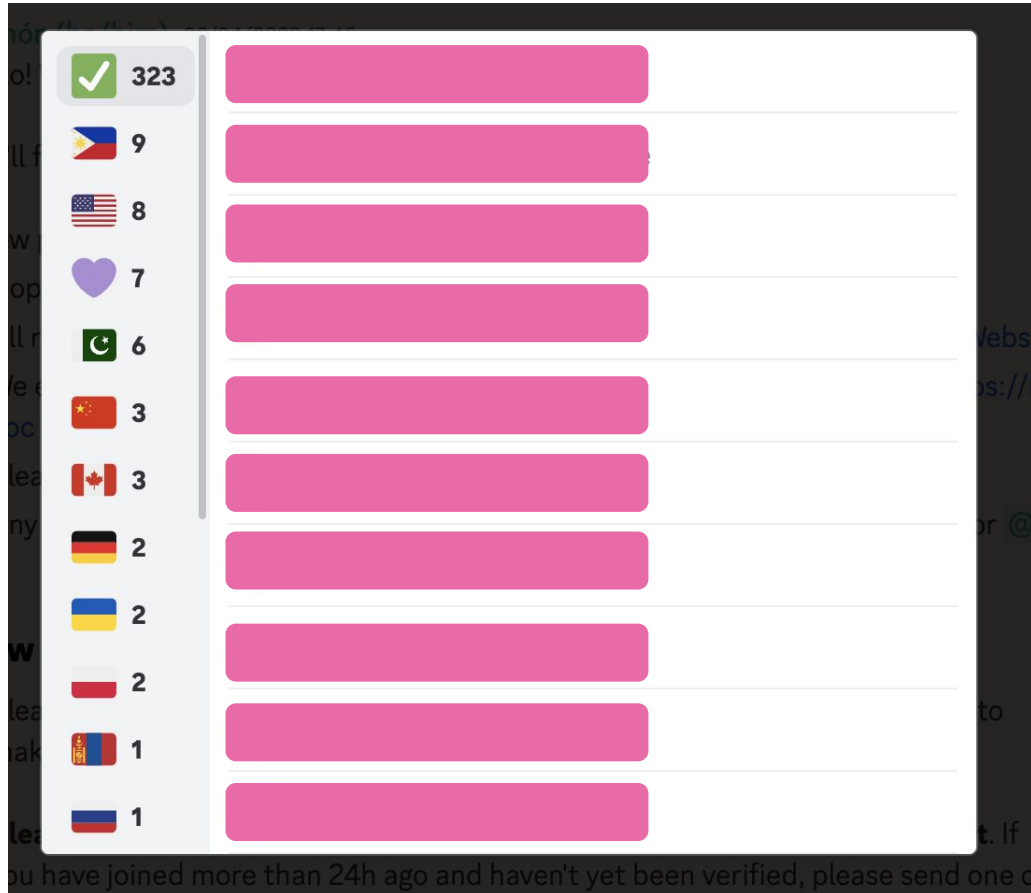
Bot Codebase



Cloud Deployment

I can now **automatically**
verify folks ✨

... but only the first 100



you have joined more than 24h ago and haven't yet been verified, please send one o

... but only the first 100



**Turns out Discord's api
has limits!**

We've got this 😊

verify.js before

```
const { SlashCommandBuilder } = require("discord.js");
const channelID = process.env.CHANNEL_ID;
const messageID = process.env.MESSAGE_ID;
const emojiID = process.env.EMOJI_ID;
const serverID = process.env.SERVER_ID;

module.exports = {
  data: new SlashCommandBuilder()
    .setName("verify")
    .setDescription("Replies with whether users are verified!"),
  async execute(client, interaction) {
    const channel = await client.channels.fetch(channelID);
    const message = await channel.messages.fetch(messageID);
    const guild = await client.guilds.fetch(serverID);
    const reaction = message.reactions.cache.get(emojiID);
    const users = await reaction.users.fetch();
```

verify.js

```
const { SlashCommandBuilder } = require("discord.js");
const channelID = process.env.CHANNEL_ID;
const messageID = process.env.MESSAGE_ID;
const emojiID = process.env.EMOJI_ID;
const serverID = process.env.SERVER_ID;

module.exports = {
  data: new SlashCommandBuilder()
    .setName("verify")
    .setDescription("Replies with whether users are verified!"),
  async execute(client, interaction) {
    const channel = await client.channels.fetch(channelID);
    const message = await channel.messages.fetch(messageID);
    const guild = await client.guilds.fetch(serverID);
    const reaction = message.reactions.cache.get(emojiID);
    const users = await reaction.users.fetch();
```

verify.js after

```
module.exports = {
  data: new SlashCommandBuilder()
    .setName("verify")
    .setDescription("Replies with whether users are verified!"),
  async execute(client, interaction) {
    const channel = await client.channels.fetch(channelID);
    const message = await channel.messages.fetch(messageID);
    const guild = await client.guilds.fetch(serverID);
    const reaction = message.reactions.cache.get(emojiID);

    const users = await fetchAllUsers(reaction);
```

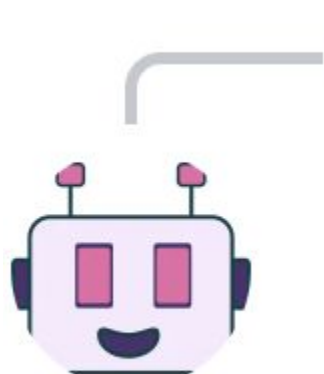
verify.js

```
const fetchAllUsers = async (reaction) => {
  const users = [];
  let last_id;

  while (true) {
    const options = { limit: 100 };
    if (last_id) {
      options.after = last_id;
    }
    const newUsers = await reaction.users.fetch(options);
    users.push(...Array.from(newUsers));
    last_id = newUsers.last().id;

    if (newUsers.size !== 100 || users.length >= 100) {
      break;
    }
  }
  return users.flat(Infinity);
};
```


Back in Discord...

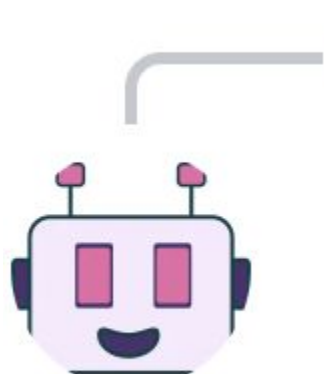


 Ramón (he/him) used `/verify`

verify-bot BOT Today at 00:22

Verifying members

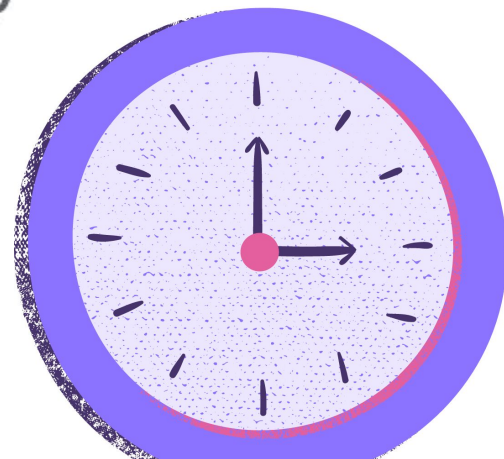
Back in Discord...



Ramón (he/him) used `/verify`

verify-bot **BOT** Today at 00:22

Verifying members



But hey, it works! 

**There's so much more to
the Discord api!**

**There's loads of events
you can catch, like
messages, reactions, etc!**

**fly.io's monitoring tools
make console.log debugging
super doable!**

**The typical case of
spending 8 hours of
saving myself 10 minutes**

**Testing is super doable,
and don't fear the bugs!**

**Y'know what, I'm feeling
dangerous...**

Actually, about that... 

Oh also...

**Turns out Discord offers
this verification out of the
box 😅**

**This is pretty
challenging, but I'm
having a blast!**

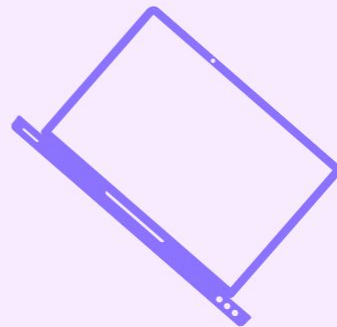
**And keeping my
community safe and
engaged 😊**

Resources:

- DiscordJS : <https://discordjs.guide/>
- Fly.io: <https://fly.io/>
- Discord Developer Portal: <https://discord.com/developers/>



Ramón Huidobro



**Thank you,
Friends!**



ramonh.dev/card

