

Criando bot do discord com o nodejs
e interagindo com bot do telegram

Applications


New Application



Find the perfect feature set for your game in our Game SDK, and sign up for [Serverless 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



You don't have any applications yet, sad face.

[Click New Application above to get started.](#)

<https://discord.com/developers/applications>

Aqui damos um nome a nossa
nova aplicação

CREATE AN APPLICATION

Are you a game dev? We may already have your app in our database. Reach out to our **Dev Support** for more info and to claim your game!

NAME *

By creating an API application, you agree to the [Discord API Terms of Service](#)

CancelCreate

Com a aplicação criada vamos até a guia de bot e criaremos um novo bot no botão **add bot**

← Back to Applications

SELECTED APP

nerd-friday

SETTINGS

General Information

OAuth2

Bot


Bot

Bring your app to life on Discord with a Bot user. Be a part of chat in your users' servers and interact with them directly.

[Learn more about bot users](#)

Build-A-Bot

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



Add Bot

Com o bot criado teremos nosso token, o token é como se fosse a identidade do nosso bot, então quem possuir o token do seu bot pode fazer o login com seu bot e usar o mesmo para qualquer aplicação.

ICON



USERNAME

nerd-friday

#9568

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.

OTk0OTY2NTcyNDIwNTgzNDI0.GGfitS.GnO_OmEiwsOprxZ5PCWuO-5c2pjUA-4VtCfgJA

Copy

Reset Token

Com o token em mãos vamos gerar nossa URL de convite para o bot, vá em **OAuth2** e depois na guia de **URL Generator**, após isso vamos selecionar a opção **bot** em **scopes**

The screenshot displays the Discord Developer Portal interface. On the left sidebar, under the 'SETTINGS' section, the 'OAuth2' option is selected and highlighted in blue. Below it, the 'URL Generator' sub-option is highlighted with a red rectangular border. The main content area on the right is titled 'SCOPES' and contains a grid of checkboxes for various permissions. The 'bot' checkbox is checked, while all other checkboxes are unchecked.

SELECTED APP

nerd-friday

SETTINGS

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

SCOPES

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

Aqui daremos as permissões que nosso bot precisa, estou dando permissões básicas ao meu bot

GENERAL PERMISSIONS

- ☐ Administrator
- ☐ View Audit Log
- ☐ Manage Server
- ☐ Manage Roles
- ☐ Manage Channels
- ☐ Kick Members
- ☐ Ban Members
- ☐ Create Instant Invite
- ☐ Change Nickname
- ☐ Manage Nicknames
- ☐ Manage Emojis and Stickers
- ☐ Manage Webhooks
- ☒ Read Messages/View Channels
- ☐ Manage Events
- ☐ Moderate Members

TEXT PERMISSIONS

- ☒ Send Messages
- ☐ Create Public Threads
- ☐ Create Private Threads
- ☐ Send Messages in Threads
- ☐ Send TTS Messages
- ☒ Manage Messages
- ☐ Manage Threads
- ☒ Embed Links
- ☒ Attach Files
- ☒ Read Message History
- ☒ Mention Everyone
- ☒ Use External Emojis
- ☐ Use External Stickers
- ☒ Add Reactions
- ☐ Use Slash Commands

VOICE PERMISSIONS

- ☐ Connect
- ☐ Speak
- ☐ Mute Members
- ☐ Deafen Members
- ☐ Move Members
- ☐ Use Voice Activity
- ☐ Priority Speaker

Por fim teremos nossa URL gerada e com ela podemos adicionar o bot ao nosso canal no discord.

Cole a URL no navegador, selecione um servidor para o qual o bot ficará e clique em Authorize

GENERATED URL

https://discord.com/api/oauth2/authorize?client_id=994966572420583424&permissions=519232&scope=bot

Copy

Para criarmos um bot no telegram precisamos usar o **BotFather** do próprio telegram, através dele criamos nosso bot.

Após criarmos o bot é nos dado um token do mesmo que será usado no projeto.

Para podermos comunicar com o bot através do **node** precisamos já ter iniciado uma conversa com bot antes.

Precisamos também do id de quem iremos enviar as mensagens, o id é pego pelo link. https://api.telegram.org/bot<token_bot>/getUpdates

- `npm init -y`
 - iniciando o `package.json`
- `npm install --save dotenv-safe`
 - Dotenv-safe nos permite trabalhar com variáveis de ambiente em nosso projeto
 - Na inicialização da aplicação é verificado se as variáveis de ambiente do `.env` estão de acordo com as do `.env.example`
- `npm install --save discord.js`
 - Módulo que permite usar a api do discord no node
- `npm install --save telegraf`
 - Módulo que permite usar a api do telegram no node

Package.json após ser iniciado com o npm init -y

```
package.json
1 {
2   "name": "nerd-friday",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC"
12 }
```

Package.json após instalação dos módulos

```
package.json
1 {
2   "name": "nerd-friday",
3   "version": "1.0.0",
4   "description": "",
5   "type": "module",
6   "main": "index.js",
7   "scripts": {
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "discord.js": "^13.8.1",
15    "dotenv-safe": "^8.2.0",
16    "telegraf": "^4.8.5"
17  }
18 }
```

Criando as variáveis de ambiente

Iremos criar dois arquivos o `.env` que conterá nossas variáveis de ambiente e o `.env.example` que terá apenas o nome das nossas variáveis para assim o `dotenv-safe` poder fazer suas validações.

`.env`

```
1 DISCORD_TOKEN=1111111111111111
2 DISCORD_OPS_CHANNEL_ID=222222222222
3 BOT_TOKEN=3333333333333333
4 CHAT_ID=444444444444
5
```

`.env.example`

```
1 DISCORD_TOKEN=
2 DISCORD_OPS_CHANNEL_ID=
3 BOT_TOKEN=
4 CHAT_ID=
```

