

Creating a Discord Bot in 5 Minutes

Creating a discord bot to interact with you and your friends is extremely easy. With no programming experience it is only a few simple steps to get up and running, and with some practice, you can have your bots do some pretty cool things like: searching for and playing music, getting information from the internet, managing your channel's members, and more. In this guide, we'll show you how to get a basic bot up and running, which you can use as a starting point for something more advanced.

In this guide we will:

1. Install Node.js and Discord.js
2. Get our bots credentials from Discord
3. Write a command with a message response

Let's start!

The first thing you'll need is a software called Node. You can download it here:

<https://nodejs.org/en/download/>

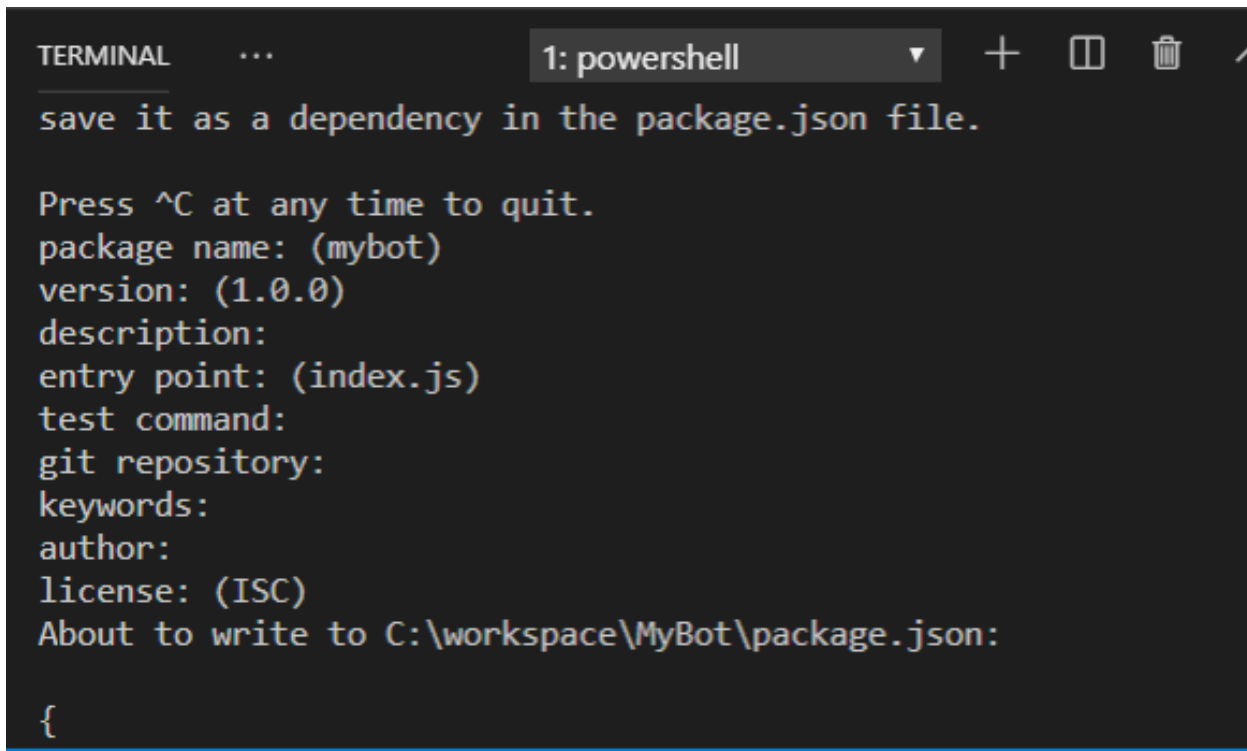
Once the download is complete, run and install the software.

Make a folder somewhere on your computer where you want the bot to live (this is where you will put your code) . ex. Documents/MyBot

Open a terminal inside the folder you created (this is where we run “commands” to get our bot started). On Windows, click the Start menu and search for **cmd** to open a terminal.

In the terminal you opened, type in and run the following:

```
npm init
```



```
TERMINAL  ...  1: powershell
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (mybot)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\workspace\MyBot\package.json:

{
```

You will be prompted to enter some information about the application, here you can enter whatever information you wish or leave them all blank (just hit **Enter** until it is done).

When that is finished:

On Linux/Mac, in your terminal, type in and run the following:

```
touch index.js
```

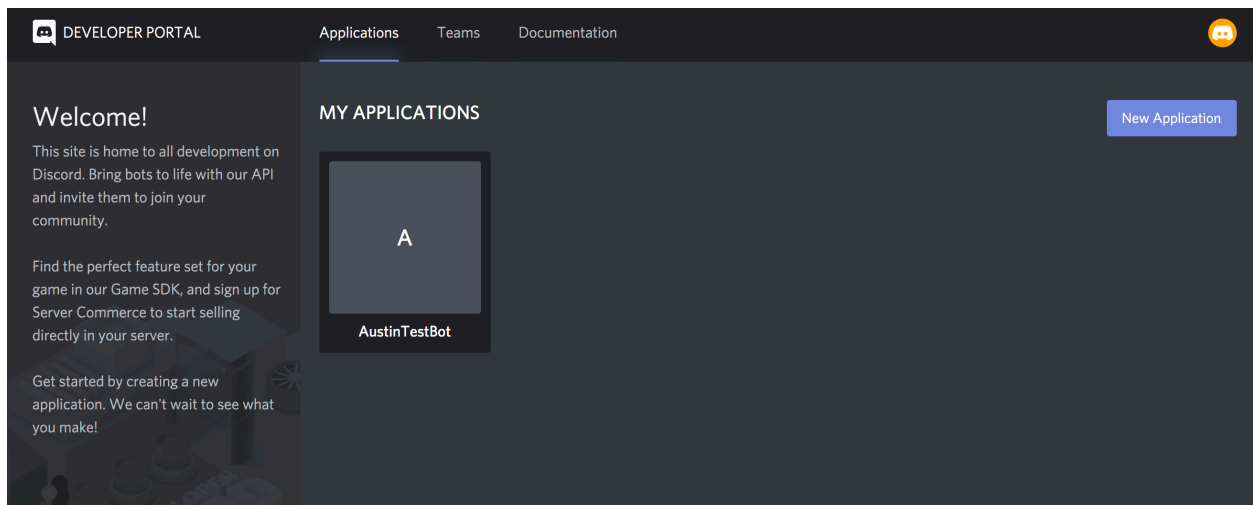
On Windows, you must ‘right-click’ and in the menu, hover over **New** and select **Text Document**. Name the file **index.js**.

Once you've created the file, type in and run the following in your terminal:

```
npm install discord.js
```

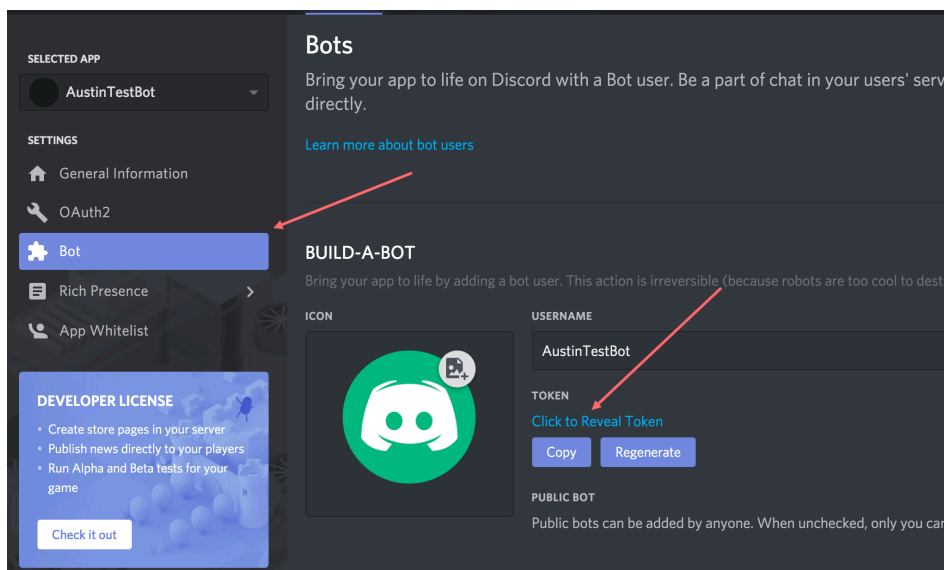
Next we want to get the credentials, also called a “token” from Discord. This “token” tells Discord who your bot is and what it is allowed to do. We will need it for our bot to work. You can find that here:

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



You will need a free Discord account for this step. Once you have one, under **Applications** you want to select **New Application**.

Once your application is created, select **Bot** in the left-hand side menu and then select **Add Bot** to create your bot and get your token. Click **Copy** under the Token section.



Now we can paste the following into our index.js file by opening the file in Notepad (on Windows) or TextEdit (on Mac) and replace “<token>” with the token you found in the previous step:

```
const Discord = require('discord.js');
const client = new Discord.Client();
client.on('ready', () => {
  console.log(`Logged in as ${client.user.tag}!`);
});
client.on('message', msg => {
  if(msg.author.bot) return;
  if (msg.content === 'ping') {
    msg.reply('Pong!');
  }
});
client.login("<token>");
```

Let's take a look at the code above, which is straight from the discord.js website.

In the code, “**client**” is how we tell our bot to listen to certain things in our Discord chatroom. What we have in the code right now says to the bot to listen for two “events”; the first is when the bot is ready, and the second is whenever a message is sent in the chatroom. Going deeper, we tell the bot to check any messages sent, and if they have the word “ping” in them, for the bot to respond with “Pong!”.

So let's test this out!

Back in your terminal, type in and run the following:

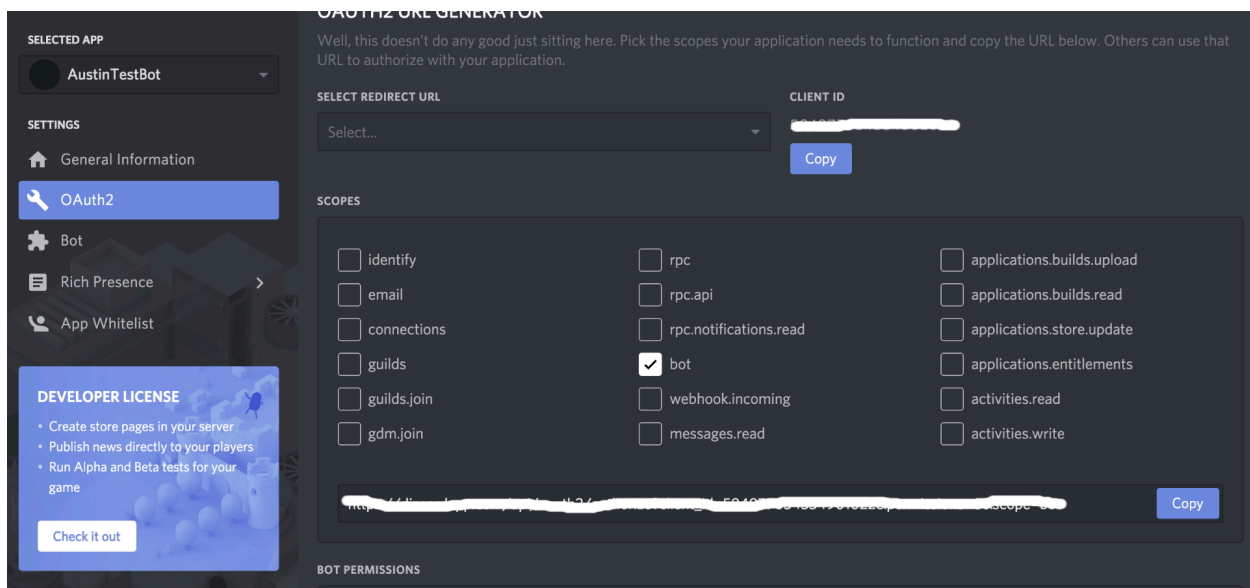
```
node index.js
```

This tells the bot to run and if it is successful, you should see a message saying that the bot is logged in.

```
MacBook-Pro-21:wwwroot Austin$ node index.js
Logged in as AustinTestBot#7622!
```

Now we need to invite the bot to our Discord server!

Back on the Discord developer website (where we got our bot's token), select the **OAuth2** option in the left-hand side menu.



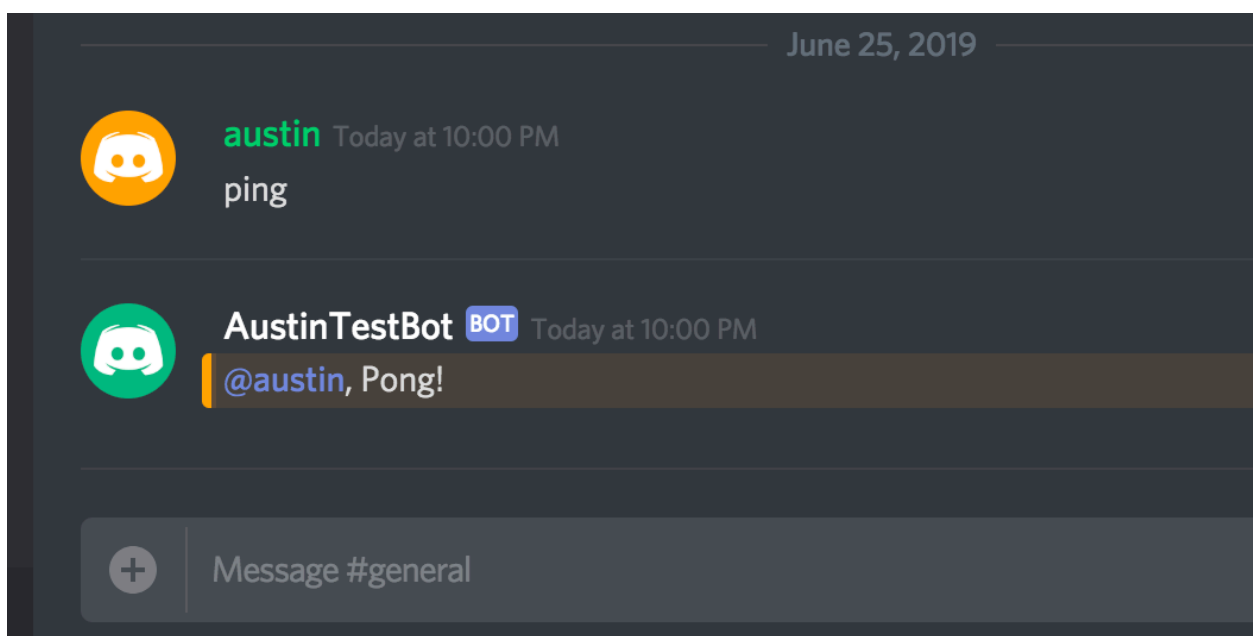
Under **Scopes**, you want to check the box that says **bot**, then scroll down to set the permissions we want the bot to have.

BOT PERMISSIONS

GENERAL PERMISSIONS	TEXT PERMISSIONS	VOICE PERMISSIONS
<input type="checkbox"/> Administrator	<input checked="" type="checkbox"/> Send Messages	<input type="checkbox"/> Connect
<input type="checkbox"/> View Audit Log	<input type="checkbox"/> Send TTS Messages	<input type="checkbox"/> Speak
<input type="checkbox"/> View Server Insights	<input type="checkbox"/> Manage Messages	<input type="checkbox"/> Mute Members
<input type="checkbox"/> Manage Server	<input type="checkbox"/> Embed Links	<input type="checkbox"/> Deafen Members
<input type="checkbox"/> Manage Roles	<input type="checkbox"/> Attach Files	<input type="checkbox"/> Use Members
<input type="checkbox"/> Manage Channels	<input checked="" type="checkbox"/> Read Message History	<input type="checkbox"/> Use Voice Activity
<input type="checkbox"/> Kick Members	<input type="checkbox"/> Mention Everyone	<input type="checkbox"/> Priority Speaker
<input type="checkbox"/> Ban Members	<input type="checkbox"/> Use External Emojis	
<input type="checkbox"/> Create Instant Invite	<input type="checkbox"/> Add Reactions	
<input type="checkbox"/> Change Nickname		
<input type="checkbox"/> Manage Nicknames		
<input type="checkbox"/> Manage Emojis		
<input type="checkbox"/> Manage Webhooks		
<input type="checkbox"/> View Channels		

Once you've done this, you can click the **Copy** button underneath the **Scopes** to get your invite link. Copy and paste this link into your browser and you will be prompted to select the server to which you would like to invite the bot. You will get a message confirming the bot is now a member of that chat. At this point, you should be able to go to your server and see the bot is now a member.

Now we can write our command "ping", and see the bot reply in the chat!



We now have a working bot!

If you want to add more features, it is very simple, but it will require some basic knowledge in programming. Check out this guide if you want to give it a go:

<https://gabrieltanner.org/blog/discord-music-bot>

Or take a look at this collection of bots whose code you can take a look at and reuse for your own purposes:

<https://top.gg/>

To check out the bots from the DTS team visit:

<https://github.com/MTS-STM/TakeOurKidsToWorkDay2019>