Discord in JavaScript

By Ethan, Hunter, Michael, Thomas and Julio

Presentation Walkthrough

- Discord General Overview (What is discord?)
- Discord Bot Overview (What is a bot?)
- Class Sample Exercise
- Our Demo Application

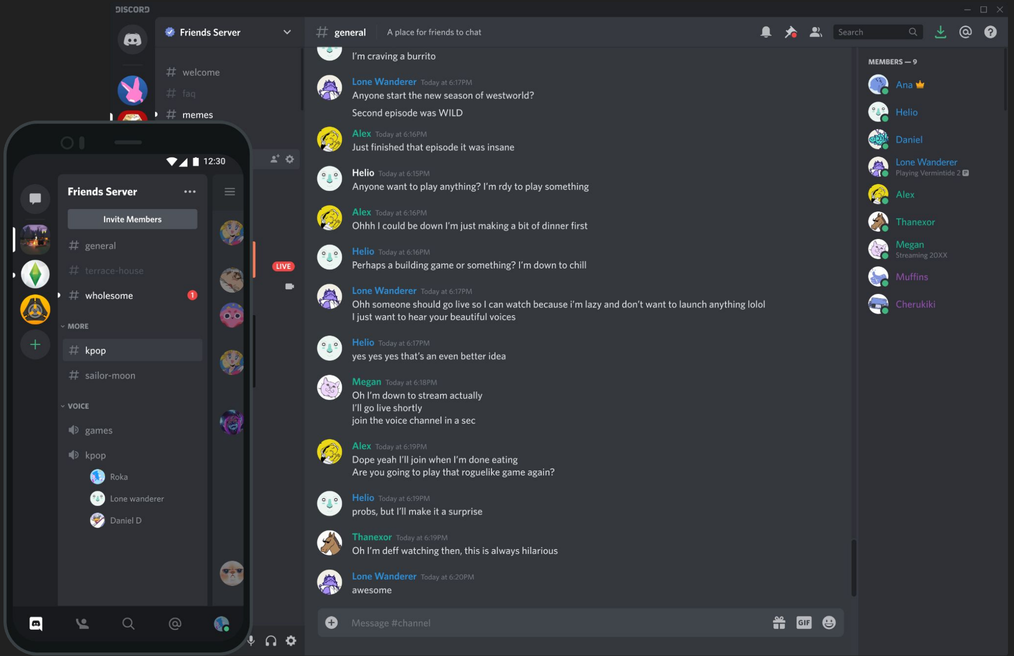
What is Discord?

- Discord is a VoIP (voice over internet protocol) that allows millions of users to connect by voice/video calls, text messages, as well as servers so communication can connect. It can also be used on almost any platform (Windows, Mac, Linux, IOS, Android, and web browsers)

How was it programmed/created?

The languages listed below are the languages that were used to created discord

- JavaScript(React)
- Python
- Elixir
- Rust
- C++



I'm craving a burrito

Lone Wanderer Today at 6:17PM
Anyone start the new season of westworld?
Second episode was WILD

Alex Today at 6:16PM
Just finished that episode it was insane

Helio Today at 6:15PM
Anyone want to play anything? I'm rdy to play something

Alex Today at 6:16PM
Ohhh I could be down I'm just making a bit of dinner first

Helio Today at 6:16PM
Perhaps a building game or something? I'm down to chill

Lone Wanderer Today at 6:17PM
Ohh someone should go live so I can watch because i'm lazy and don't want to launch anything lolol
I just want to hear your beautiful voices

Helio Today at 6:17PM
yes yes yes that's an even better idea

Megan Today at 6:18PM
Oh I'm down to stream actually
I'll go live shortly
join the voice channel in a sec

Alex Today at 6:19PM
Dope yeah I'll join when I'm done eating
Are you going to play that roguelike game again?

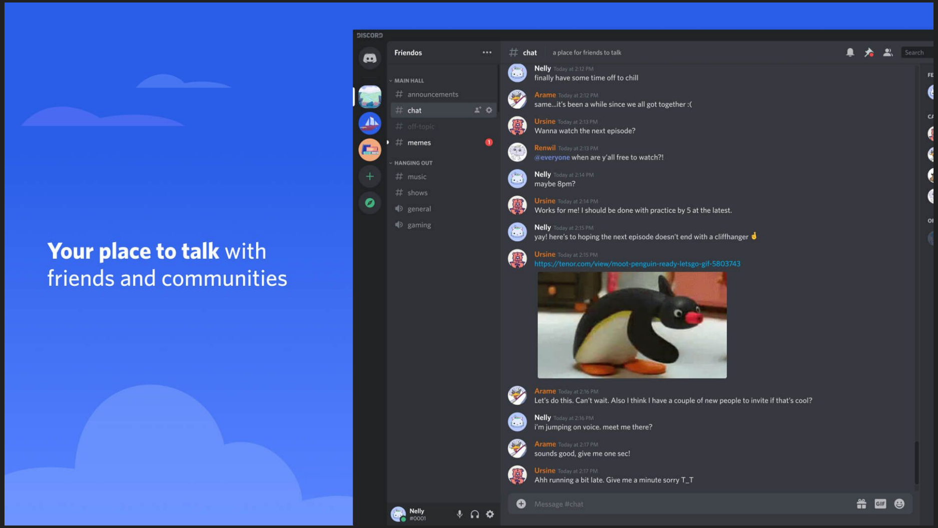
Helio Today at 6:19PM
probs, but I'll make it a surprise

Thanexor Today at 6:19PM
Oh I'm deff watching then, this is always hilarious

Lone Wanderer Today at 6:20PM
awesome

What is discord used for?

- Discord has many different uses the main purpose as I said before is for people to connect and communicate online across the world
- It can also be used in school/work environments so students/employees can communicate with each other.
- Discord also has features such as file sharing, screen sharing, gifting, music listen along session, and much more especially when it comes to bots.



Nelly Today at 2:12 PM
finally have some time off to chill

Arame Today at 2:12 PM
same...it's been a while since we all got together :(

Ursine Today at 2:13 PM
Wanna watch the next episode?

Renwil Today at 2:13 PM
@everyone when are y'all free to watch?!

Nelly Today at 2:14 PM
maybe 8pm?

Ursine Today at 2:14 PM
Works for me! I should be done with practice by 5 at the latest.

Nelly Today at 2:15 PM
yay! here's to hoping the next episode doesn't end with a cliffhanger 🤞

Ursine Today at 2:15 PM
https://tenor.com/view/moot-penguin-ready-letsgo-gif-5803743

Arame Today at 2:16 PM
Let's do this. Can't wait. Also I think I have a couple of new people to invite if that's cool?

Nelly Today at 2:16 PM
i'm jumping on voice. meet me there?

Arame Today at 2:17 PM
sounds good, give me one sec!

Ursine Today at 2:17 PM
Ahh running a bit late. Give me a minute sorry T_T

Who created discord?

- The 2 founders of discord are Jason Citron and Stan Vishnevskiy. Their goal was to solve a big problem which was “How to communicate with friends around the world while playing online games”
- Jason Citron before creating discord founded OpenFeint which is a platform for mobile games. On the other hand Stan founded Guildwork which is another social network for MMORPG players.

Guildwork

OpenFeint

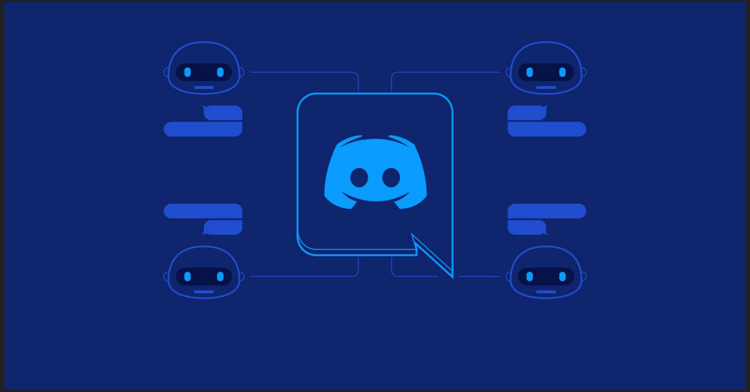
What is a Bot?

- Program that behaves like a member

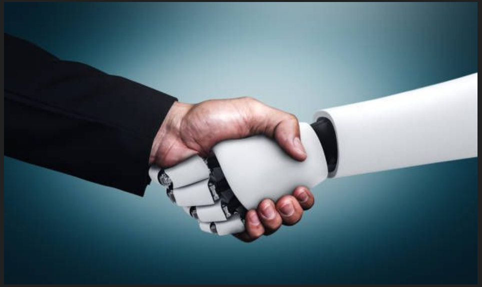
- Programmed with automatic responses

- Identified by “Bot” in its name

- Performs actions using the Discord API



What are Bots used for?



- Introduce new members

- Ban members

- Control user interaction

- Play music

- Share links / images

- Display specific info (e.g Helper Bot displays due dates)

Some popular Discord Bots

MEE6

- Welcomes new users to the server

- Bans users who violate guidelines

- Uses a loyalty level system for users to compete

- Can assign roles to users

- Sends direct messages

Some popular Discord Bots (cont.)

ProBot

- Moderation tool bot

- Fully customizable guidelines by programmer

- The programmer can decide what happens to users that violate guidelines

- Uses auto-moderation to mute users, delete spam content, etc.

Top GG

- Platform for Discord bot promotion and discovery

- Search function helps users find desired bots

- Ratings and reviews assist users in selection

- Leaderboard ranks bots by server invites

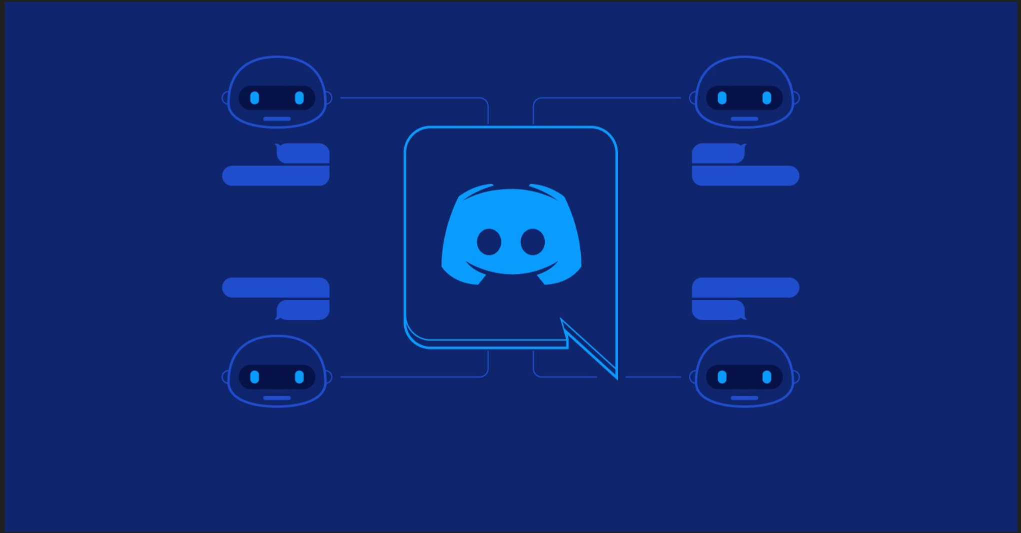
-Popular resource for bot developers and users

-Additional features include advertising and analytics

https://top.gg/servers/list/top

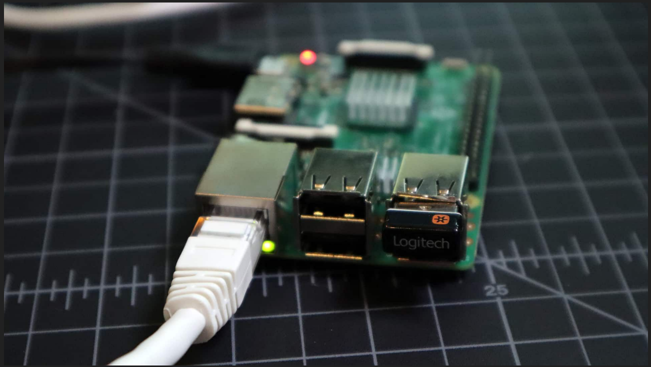
Hosting a discord bot

- A Discord bot can be very powerful and perform many tasks. So to maximize security, productivity, and more control over your bot you can host it.
- Bot hosting also ensures that the bot is up 24/7 and you won’t have to worry about lag or downtime



- 3 free hosting solutions
- Heroku
- AWS Free Tier
- Digital Ocean

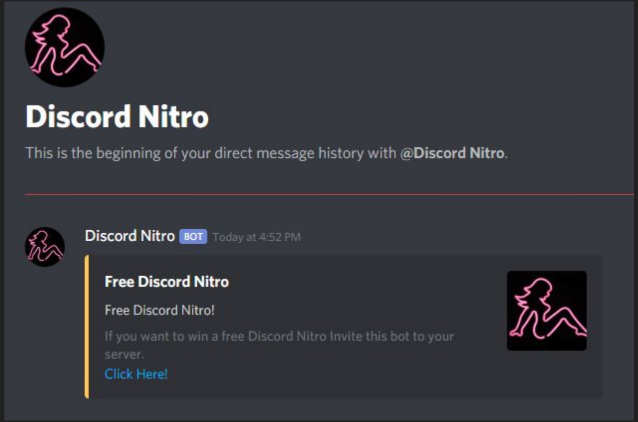
Hosting Your Discord Bot on Personal Hardware



- Choose low-power device: Raspberry Pi is ideal for energy efficiency and long-term use.Old computer would work fine as well
- Install lightweight OS: Use Raspbian Lite or Ubuntu Server for minimal resource usage.
- Set up remote access: SSH or VNC for remote management and troubleshooting.
- Use process manager: PM2 monitors bot status and automatically restarts if needed.

Security Considerations

- Malicious code: Careful development avoids harmful code; only trust verified libraries and reviewed code.
- Token leaks: Protect the bot token; leaking it allows access to the server. Keep it private and secure.
- Phishing attacks: Beware of suspicious messages asking for bot token or sensitive information.
- Permissions: Grant only necessary bot permissions to avoid abuse.
- Third-party integrations: Review third-party integrations and security policies to ensure trustworthiness



Discord Nitro

This is the beginning of your direct message history with **@Discord Nitro**.

Discord Nitro BOT Today at 4:52 PM

Free Discord Nitro

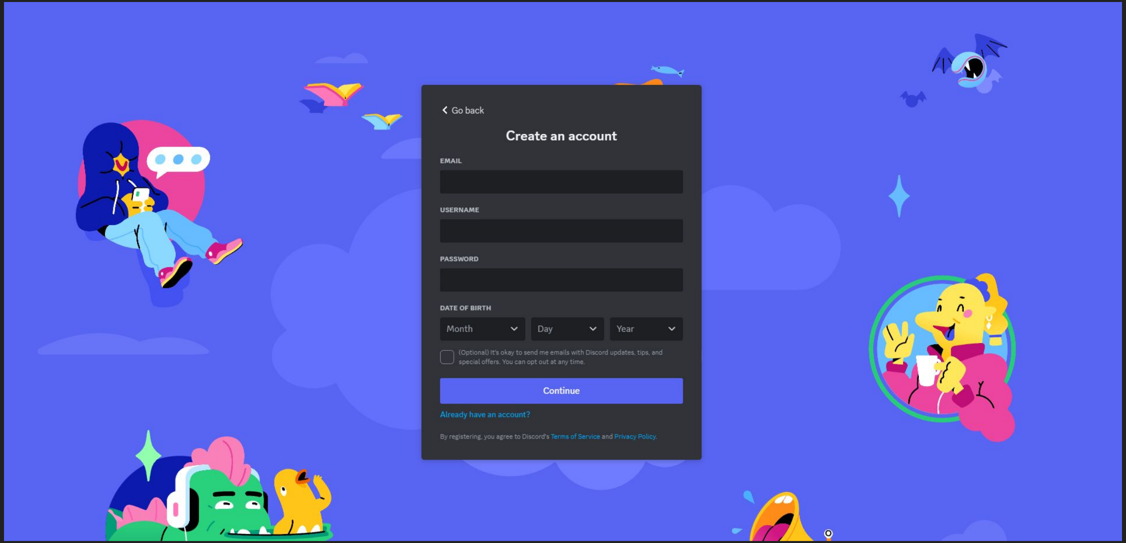
Free Discord Nitro!

If you want to win a free Discord Nitro Invite this bot to your server.

Click Here!

Sample Exercise - Creating Discord account(If applicable)

- To register for a discord account just click here
- Follow the instructions to create the account

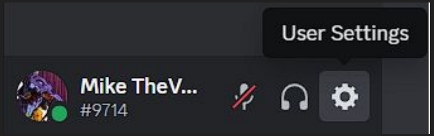


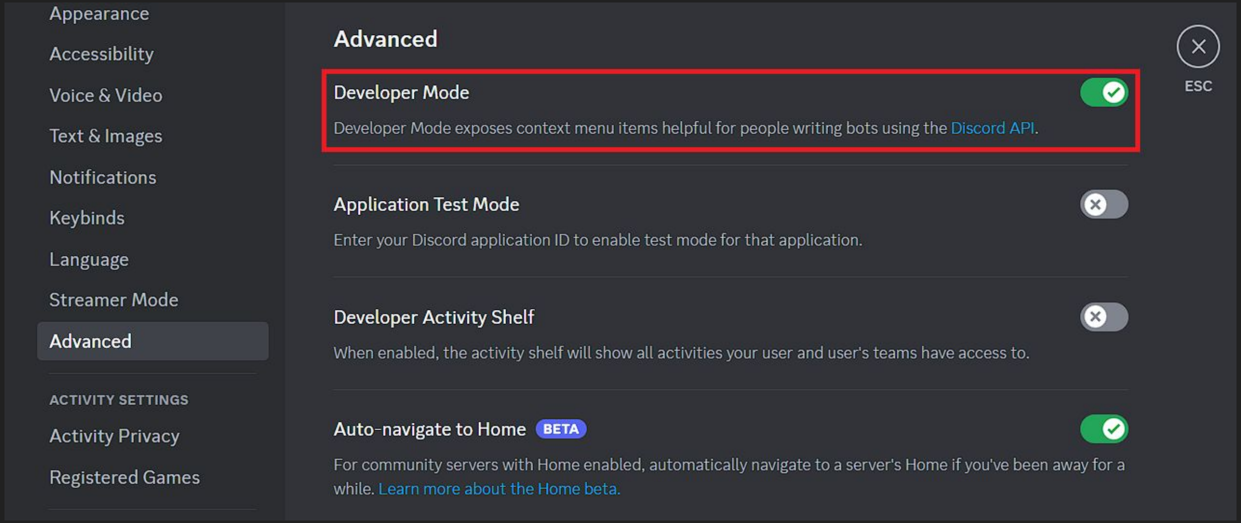
(Optional) It's okay to send me emails with Discord updates, tips, and special offers. You can opt out at any time.

By registering, you agree to Discord's Terms of Service and Privacy Policy.

Sample Exercise - Discord setup

- Turn developer mode on for your Discord account





Advanced

Developer Mode ✓

Developer Mode exposes context menu items helpful for people writing bots using the Discord API.

Application Test Mode ✗

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

Developer Activity Shelf ✗

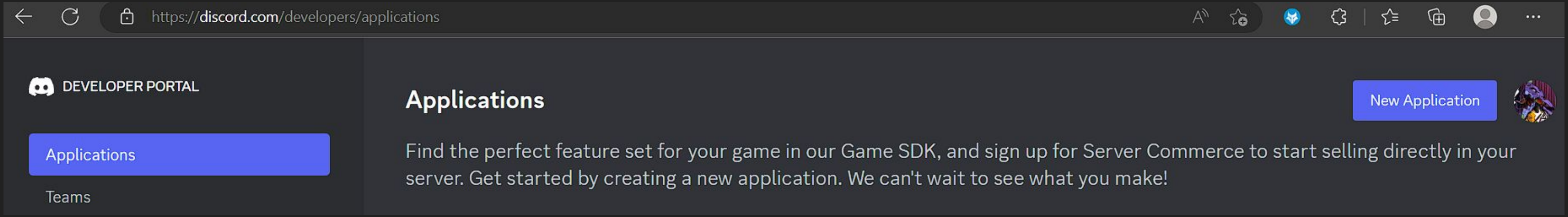
When enabled, the activity shelf will show all activities your user and user's teams have access to.

Auto-navigate to Home BETA ✓

For community servers with Home enabled, automatically navigate to a server's Home if you've been away for a while. Learn more about the Home beta.

Sample Exercise - Discord setup (cont.)

- Go to the Discord developer portal (https://discord.com/developers) and create a new application

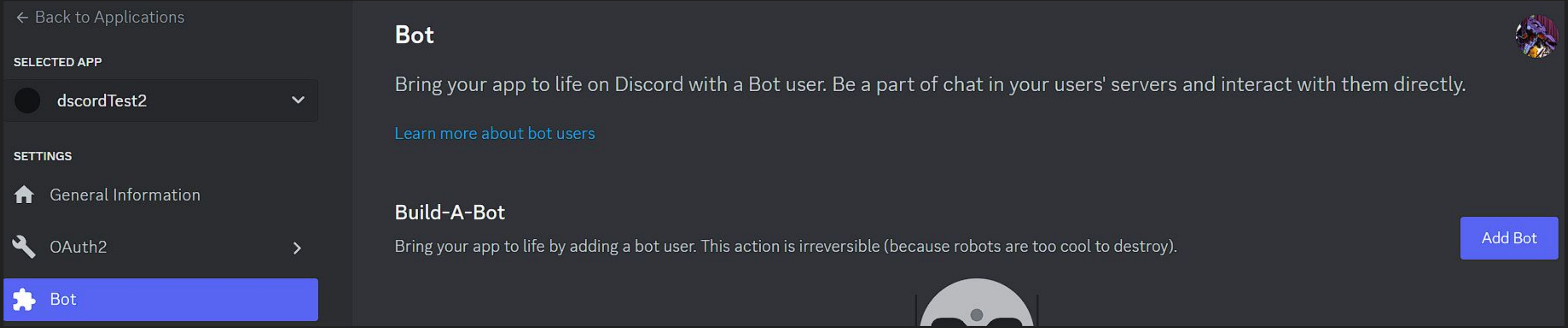


Applications

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!

Sample Exercise - Discord setup (cont.)

- Select your new application, go to the "Bot" section, and add a bot to the app



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.

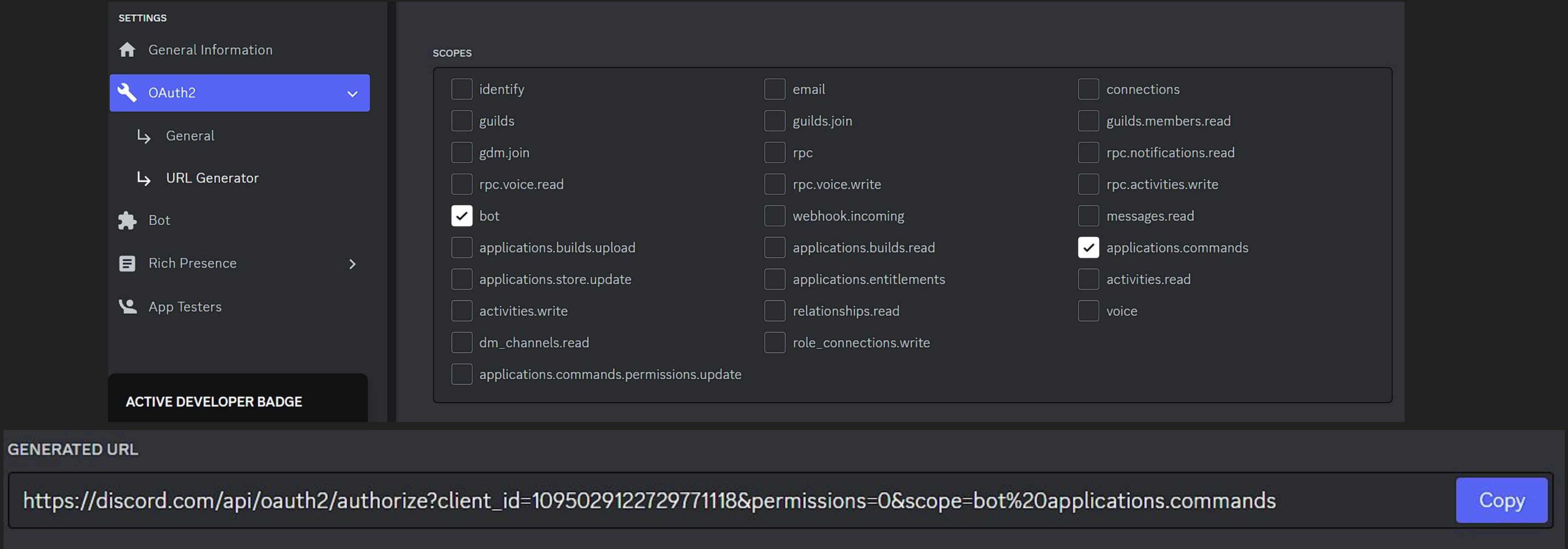
Learn more about bot users

Build-A-Bot

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

Sample Exercise - Discord setup (cont.)

- Use the OAuth2 URL Generator to create an invite link for your application, selecting the appropriate options (For this app, just use bot & applications.command)



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

ACTIVE DEVELOPER BADGE

SCOPES

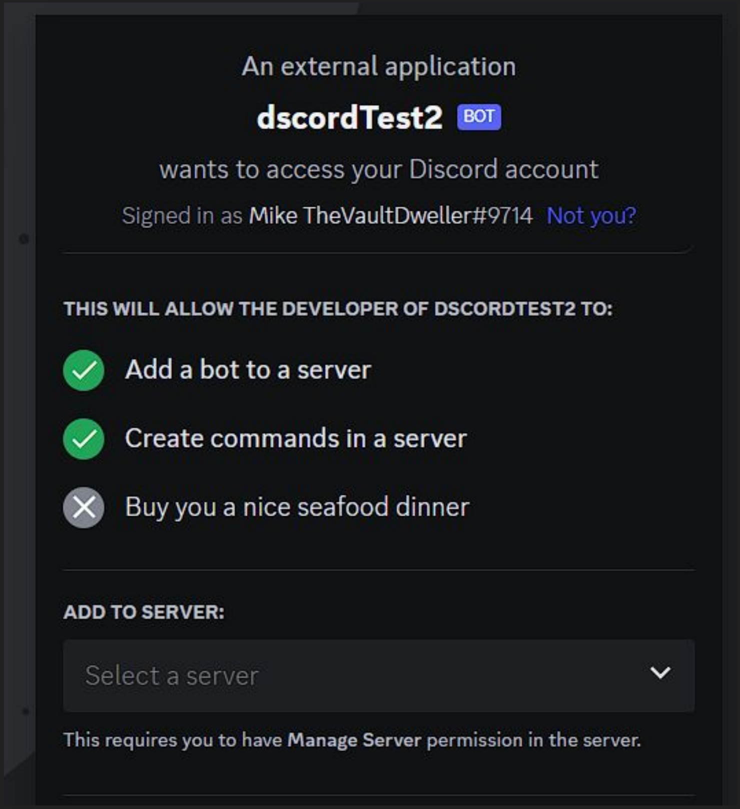
☐ identify	☐ email	☐ connections
☐ guilds	☐ guilds.join	☐ guilds.members.read
☐ gdm.join	☐ rpc	☐ rpc.notifications.read
☐ rpc.voice.read	☐ rpc.voice.write	☐ rpc.activities.write
✓ bot	☐ webhook.incoming	☐ messages.read
☐ applications.builds.upload	☐ applications.builds.read	✓ applications.commands
☐ applications.store.update	☐ applications.entitlements	☐ activities.read
☐ activities.write	☐ relationships.read	☐ voice
☐ dm_channels.read	☐ role_connections.write	
☐ applications.commands.permissions.update		

GENERATED URL

```
https://discord.com/api/oauth2/authorize?client_id=1095029122729771118&permissions=0&scope=bot%20applications.commands
```

Copy

Sample Exercise - Discord setup (cont.)



An external application

dscordTest2 BOT

wants to access your Discord account

Signed in as Mike TheVaultDweller#9714 Not you?

THIS WILL ALLOW THE DEVELOPER OF DSCORDTEST2 TO:

✓ Add a bot to a server

✓ Create commands in a server

✗ Buy you a nice seafood dinner

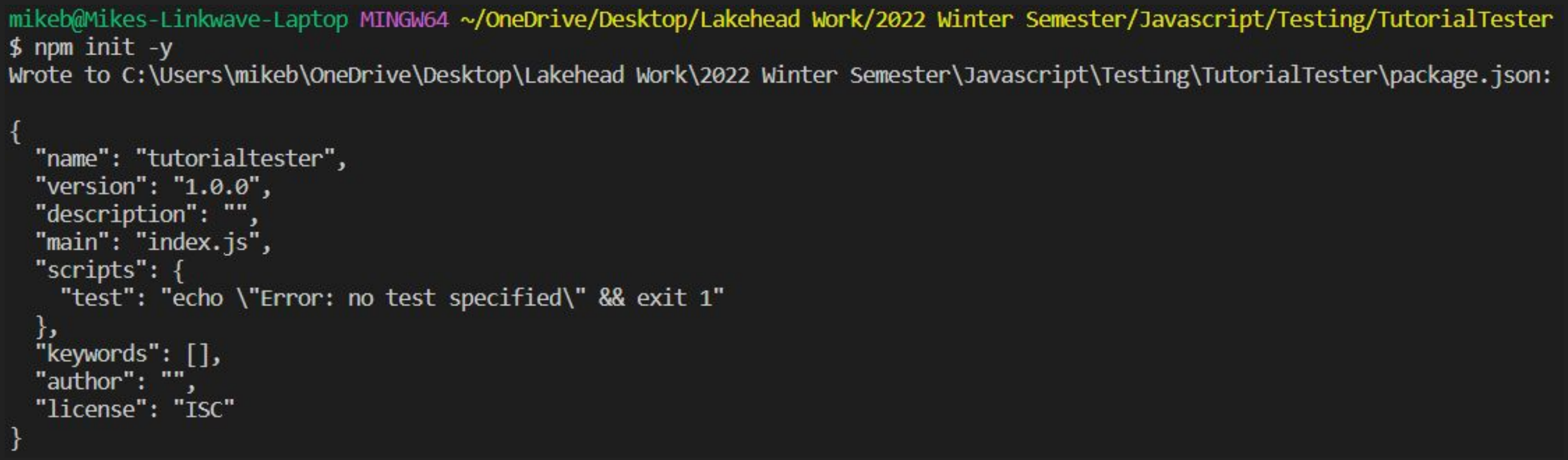
ADD TO SERVER:

Select a server

This requires you to have **Manage Server** permission in the server.

Sample Exercise - Environment setup

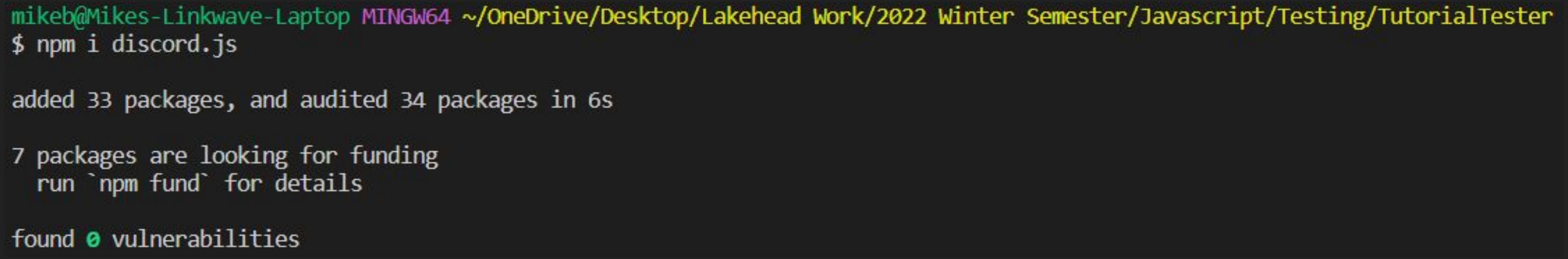
- Environment Setup - First we are going to create a Project folder and run the Command **npm init** to setup package.json file to keep track of dependencies. (-y can be added to the end to auto-generate)



```
mikeb@Mikes-Linkwave-Laptop MINGW64 ~/OneDrive/Desktop/Lakehead Work/2022 Winter Semester/Javascript/Testing/TutorialTester
$ npm init -y
Wrote to C:\Users\mikeb\OneDrive\Desktop\Lakehead Work\2022 Winter Semester\Javascript\Testing\TutorialTester\package.json:

{
  "name": "tutorialtester",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Sample Exercise - Environment setup (cont.)

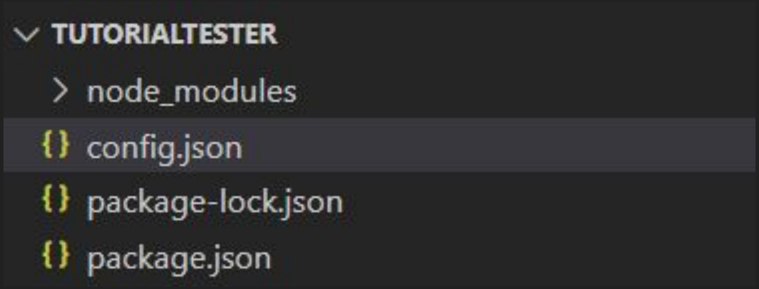
- Run **npm i discord.js** to install the dependencies for discord



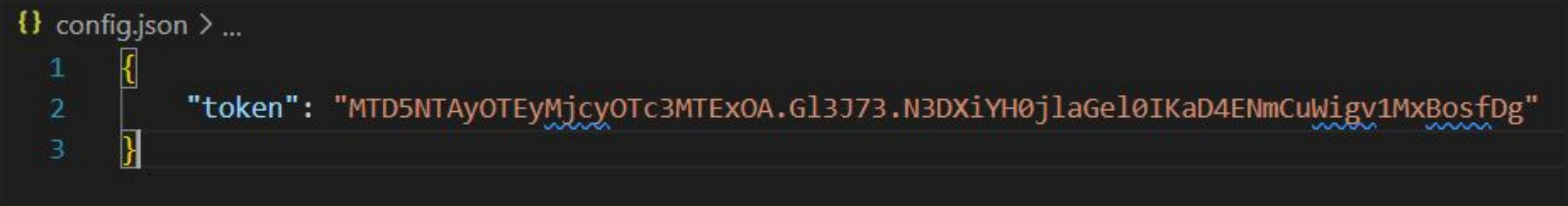
```
mikeb@Mikes-Linkwave-Laptop MINGW64 ~/OneDrive/Desktop/Lakehead Work/2022 Winter Semester/Javascript/Testing/TutorialTester
$ npm i discord.js

added 33 packages, and audited 34 packages in 6s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Sample Exercise - Bot Configuration

- Create a config.json file in the root project directory and add the key to it. **BE SURE TO ADD THIS FILE TO GITIGNORE!**



```
TUTORIALTESTER
  node_modules
  config.json
  package-lock.json
  package.json
```

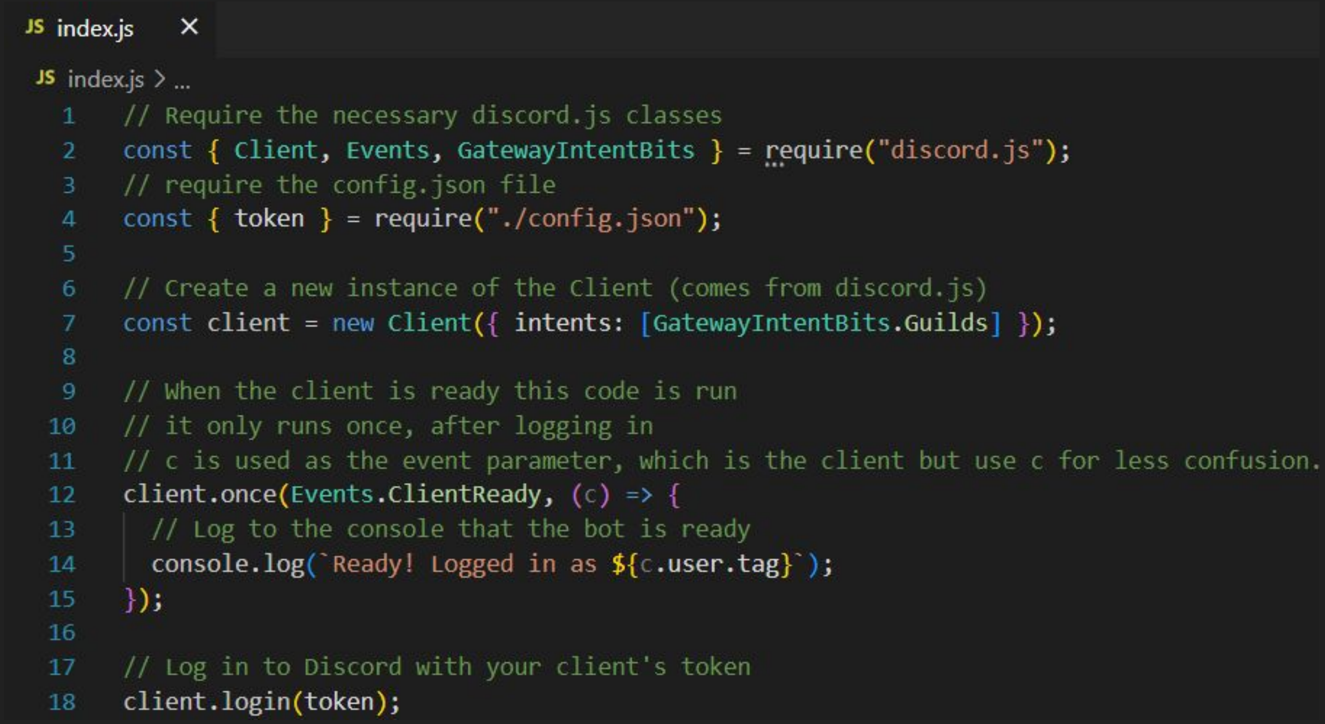


config.json > ...

```json
{
    "token": "MTD5NTAyOTEyMjcyOTc3MTExOA.Gl3J73.N3DXiYH0jlaGel0IKaD4ENmCuWigv1MxBosfDg"
}
```

Sample Exercise - Bot Configuration (cont.)

- Create an index.js file and add the following code



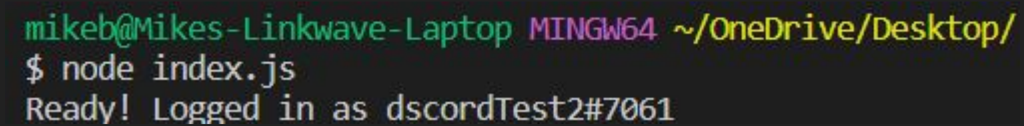
index.js ×

index.js > ...

```javascript
// Require the necessary discord.js classes
const { Client, Events, GatewayIntentBits } = require("discord.js");
// require the config.json file
const { token } = require("./config.json");

// Create a new instance of the Client (comes from discord.js)
const client = new Client({ intents: [GatewayIntentBits.Guilds] });

// When the client is ready this code is run
// it only runs once, after logging in
// c is used as the event parameter, which is the client but use c for less confusion.
client.once(Events.ClientReady, (c) => {
  // Log to the console that the bot is ready
  console.log(`Ready! Logged in as ${c.user.tag}`);
});

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

Sample Exercise - Bot Configuration (testing)

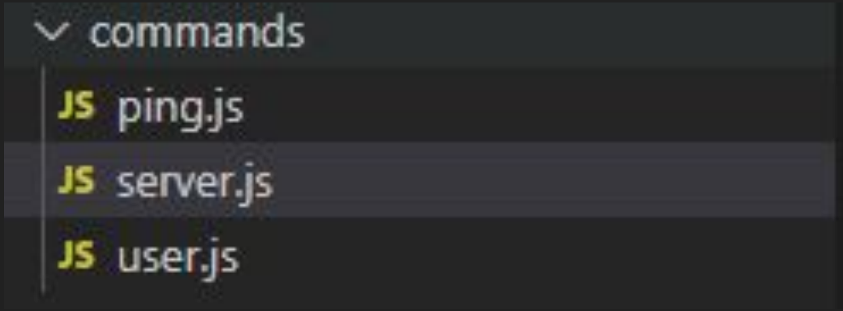
- Open the terminal and run node index.js. You should see “Ready!” in the terminal after a few seconds - this means your bot can log on to the server and is ready for the next steps!



```
mikeb@Mikes-Linkwave-Laptop MINGW64 ~/OneDrive/Desktop/
$ node index.js
Ready! Logged in as dscordTest2#7061
```

Sample Exercise - Creating Bot Slash Commands

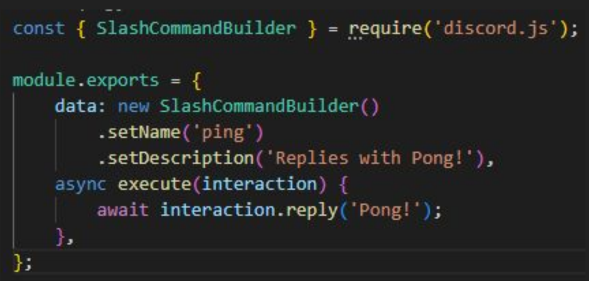
- Now we are going to create a few simple commands. To start we will create a folder called commands.
- Now create three files in the command folder ping.js, server.js, and, user.js.
- Then we will add the code to the files.



```
commands
  ping.js
  server.js
  user.js
```

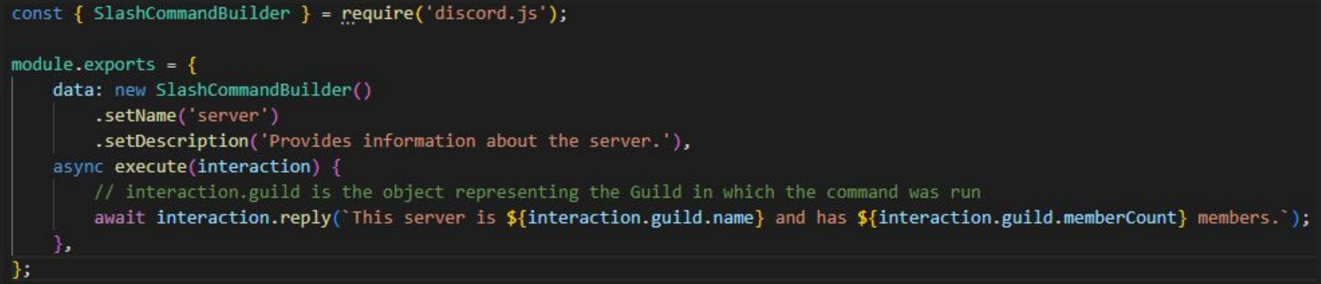
Sample Exercise - Creating Bot Slash Commands (Cont.)

ping.js



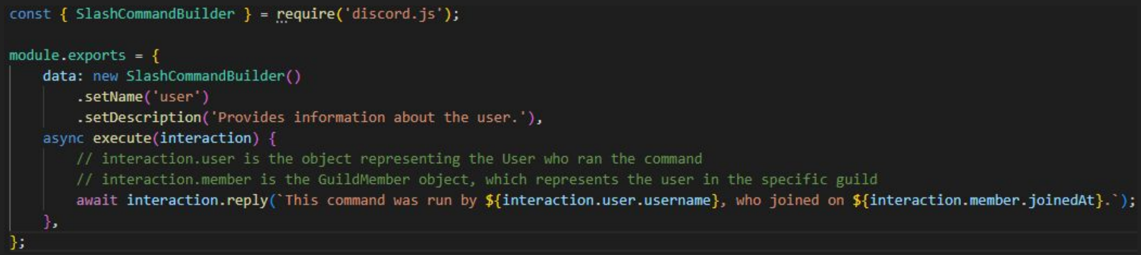
```
const { SlashCommandBuilder } = require('discord.js');

module.exports = {
    data: new SlashCommandBuilder()
        .setName('ping')
        .setDescription('Replies with Pong!'),
    async execute(interaction) {
        await interaction.reply('Pong!');
    },
};
```

Server.js



```
const { SlashCommandBuilder } = require('discord.js');

module.exports = {
    data: new SlashCommandBuilder()
        .setName('server')
        .setDescription('Provides information about the server.'),
    async execute(interaction) {
        // interaction.guild is the object representing the Guild in which the command was run
        await interaction.reply(`This server is ${interaction.guild.name} and has ${interaction.guild.memberCount} members.`);
    },
};
```

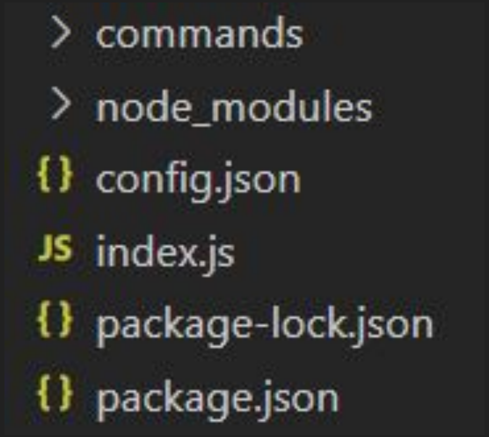
user.js



```
const { SlashCommandBuilder } = require('discord.js');

module.exports = {
    data: new SlashCommandBuilder()
        .setName('user')
        .setDescription('Provides information about the user.'),
    async execute(interaction) {
        // interaction.user is the object representing the User who ran the command
        // interaction.member is the GuildMember object, which represents the user in the specific guild
        await interaction.reply(`This command was run by ${interaction.user.username}, who joined on ${interaction.member.joinedAt}.`);
    },
};
```

Sample Exercise - Creating Bot Slash Commands (Cont.)

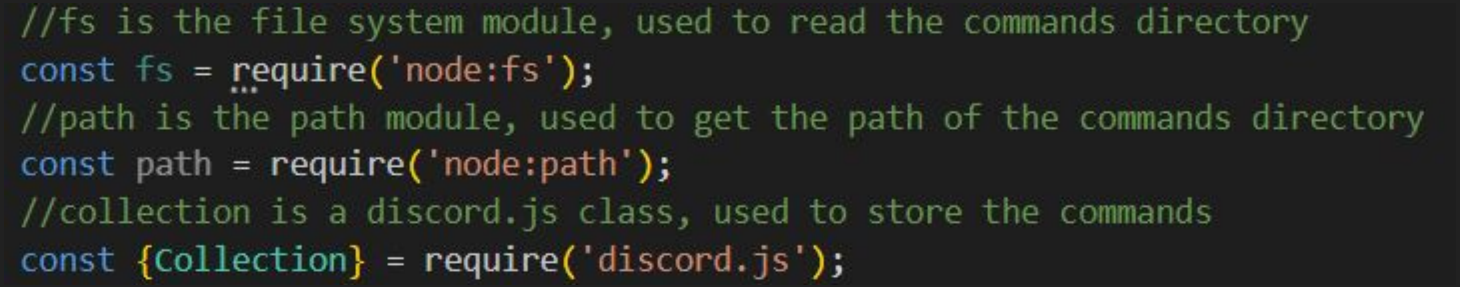
- Slash commands are an easy way for users to interact with the Bot, using commands that start with the “/” (Below is what your project should look like)



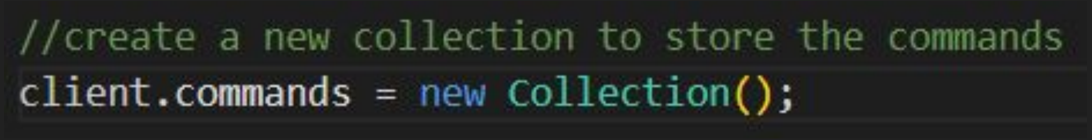
```
> commands
> node_modules
{} config.json
JS index.js
{} package-lock.json
{} package.json
```

Sample Exercise - Handling commands

- Now we need to have the bot load the command files on startup. In index.js, add the following code:



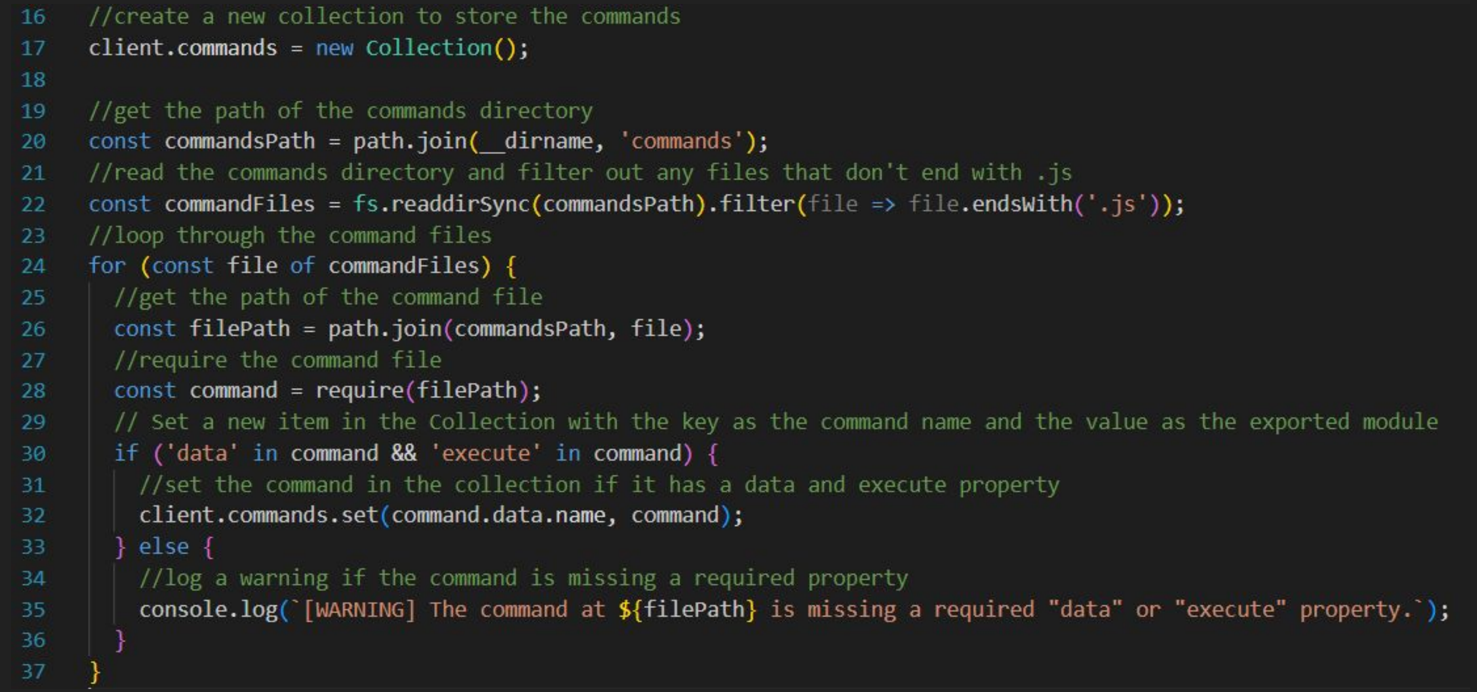
```
//fs is the file system module, used to read the commands directory
const fs = require('node:fs');
//path is the path module, used to get the path of the commands directory
const path = require('node:path');
//collection is a discord.js class, used to store the commands
const {Collection} = require('discord.js');
```



```
//create a new collection to store the commands
client.commands = new Collection();
```

Sample Exercise - Handling commands (cont.)

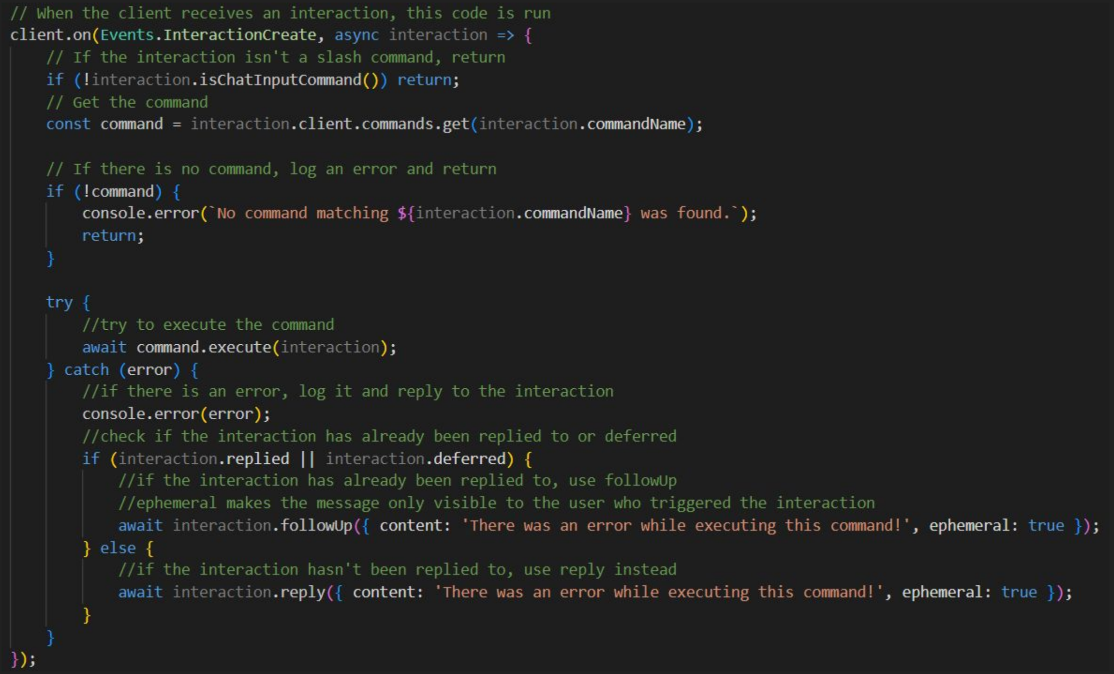
- Now we will set the bot to loop through all of the commands in the folder and save them each as a separate command (still in index.js)



```javascript
//create a new collection to store the commands
client.commands = new Collection();

//get the path of the commands directory
const commandsPath = path.join(__dirname, 'commands');
//read the commands directory and filter out any files that don't end with .js
const commandFiles = fs.readdirSync(commandsPath).filter(file => file.endsWith('.js'));
//loop through the command files
for (const file of commandFiles) {
  //get the path of the command file
  const filePath = path.join(commandsPath, file);
  //require the command 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) {
    //set the command in the collection if it has a data and execute property
    client.commands.set(command.data.name, command);
  } else {
    //log a warning if the command is missing a required property
    console.log(`[WARNING] The command at ${filePath} is missing a required "data" or "execute" property.`);
  }
}
```

Sample Exercise - Receiving slash commands

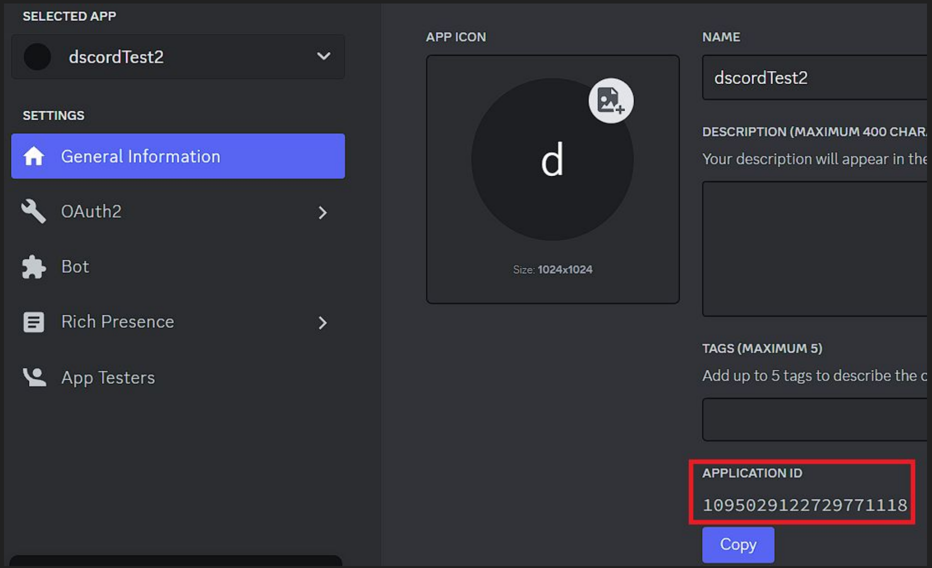
- In index.js, we will add the following code, which tells the bot how to handle slash interactions by trying execute the commands we created earlier.



```javascript
// When the client receives an interaction, this code is run
client.on(Events.InteractionCreate, async interaction => {
    // If the interaction isn't a slash command, return
    if (!interaction.isChatInputCommand()) return;
    // Get the command
    const command = interaction.client.commands.get(interaction.commandName);

    // If there is no command, log an error and return
    if (!command) {
        console.error(`No command matching ${interaction.commandName} was found.`);
        return;
    }

    try {
        //try to execute the command
        await command.execute(interaction);
    } catch (error) {
        //if there is an error, log it and reply to the interaction
        console.error(error);
        //check if the interaction has already been replied to or deferred
        if (interaction.replied || interaction.deferred) {
            //if the interaction has already been replied to, use followUp
            //ephemeral makes the message only visible to the user who triggered the interaction
            await interaction.followUp({ content: 'There was an error while executing this command!', ephemeral: true });
        } else {
            //if the interaction hasn't been replied to, use reply instead
            await interaction.reply({ content: 'There was an error while executing this command!', ephemeral: true });
        }
    }
});
```

Sample Exercise - Registering slash commands

- Before we do anything else, we need to update our config.json file to add the guildId (Server ID) and the clientId (Application ID)


SELECTED APP

dscordTest2

SETTINGS

General Information

OAuth2

Bot

Rich Presence

App Testers

APP ICON

d

Size: 1024x1024

NAME

dscordTest2

DESCRIPTION (MAXIMUM 400 CHAR

Your description will appear in the

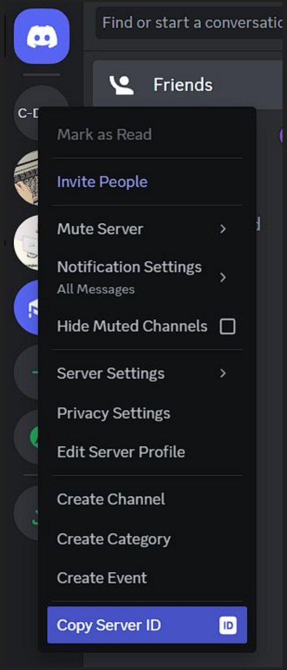
TAGS (MAXIMUM 5)

Add up to 5 tags to describe the c

APPLICATION ID

1095029122729771118

Copy


Find or start a conversatic

Friends

Mark as Read

Invite People

Mute Server

Notification Settings

All Messages

Hide Muted Channels

Server Settings

Privacy Settings

Edit Server Profile

Create Channel

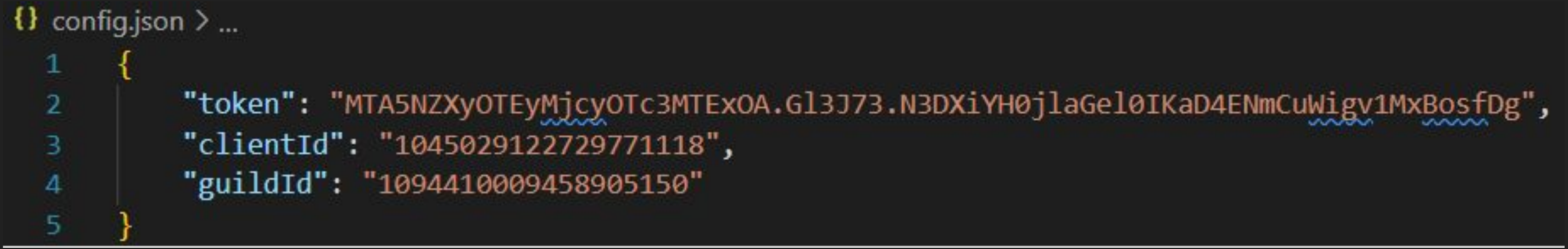
Create Category

Create Event

Copy Server ID

Sample Exercise - Registering slash commands (cont.)

- Our config.json file should now look like this

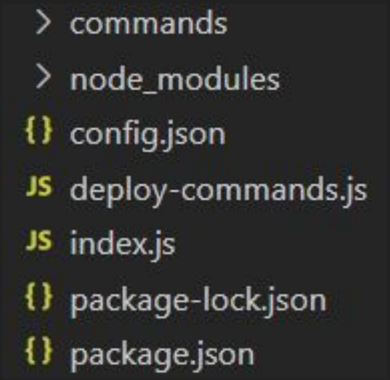


config.json > ...

```json
{
    "token": "MTA5NZXyOTEyMjcyOTc3MTExOA.Gl3J73.N3DXiYH0jlaGel0IKaD4ENmCuWigv1MxBosfDg",
    "clientId": "1045029122729771118",
    "guildId": "1094410009458905150"
}
```

Sample Exercise - Registering slash commands (cont.)

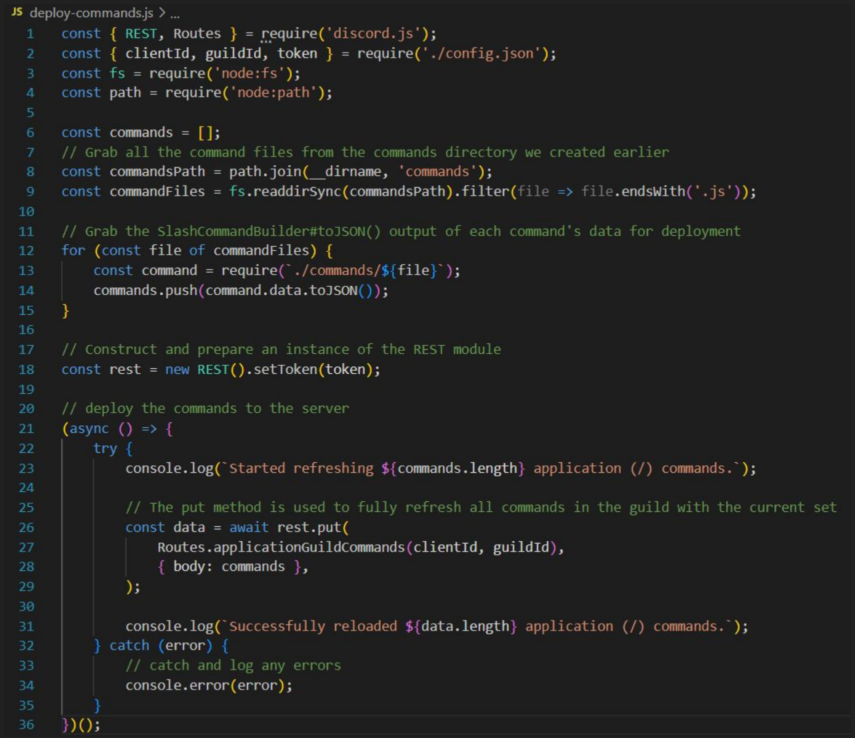
- Now we register the slash commands with Discord. These only need to be registered once, and then updated if the definition (description, options, etc.) changes.
- Discord limits the number of registrations a day, so we will create a separate script to handle registering commands.
- Create a new file in the root directory, and call it deploy-commands.js



```
> commands
> node_modules
{} config.json
JS deploy-commands.js
JS index.js
{} package-lock.json
{} package.json
```

Sample Exercise - Registering slash commands (cont.)

- Now we can write the script for deploy-commands.js

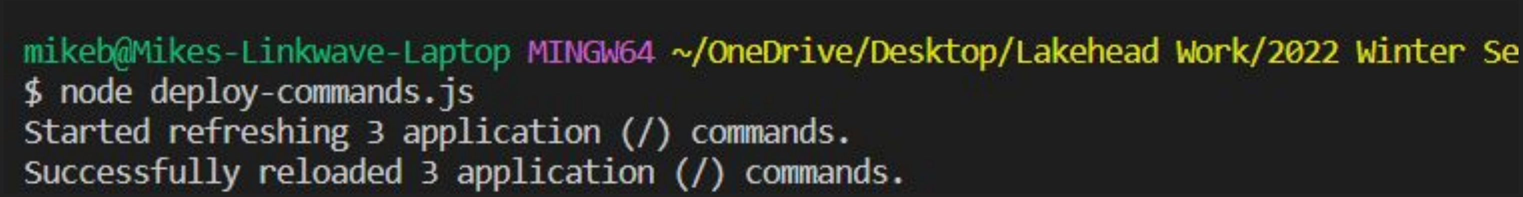


JS deploy-commands.js > ...

```javascript
const { REST, Routes } = require('discord.js');
const { clientId, guildId, token } = require('./config.json');
const fs = require('node:fs');
const path = require('node:path');

const commands = [];
// Grab all the command files from the commands directory we created earlier
const commandsPath = path.join(__dirname, 'commands');
const commandFiles = fs.readdirSync(commandsPath).filter(file => file.endsWith('.js'));

// Grab the SlashCommandBuilder#toJSON() output of each command's data for deployment
for (const file of commandFiles) {
    const command = require(`./commands/${file}`);
    commands.push(command.data.toJSON());
}

// Construct and prepare an instance of the REST module
const rest = new REST().setToken(token);

// deploy the commands to the server
(async () => {
    try {
        console.log(`Started refreshing ${commands.length} application (/) commands.`);

        // The put method is used to fully refresh all commands in the guild with the current set
        const data = await rest.put(
            Routes.applicationGuildCommands(clientId, guildId),
            { body: commands },
        );

        console.log(`Successfully reloaded ${data.length} application (/) commands.`);
    } catch (error) {
        // catch and log any errors
        console.error(error);
    }
})();
```

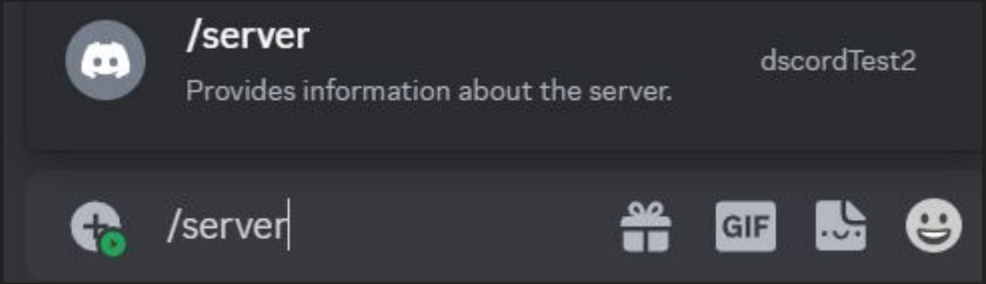
Sample Exercise - Registering slash commands (cont.)

- With our script finished, we can now run node deploy-commands.js in a new terminal to upload the commands to our server.

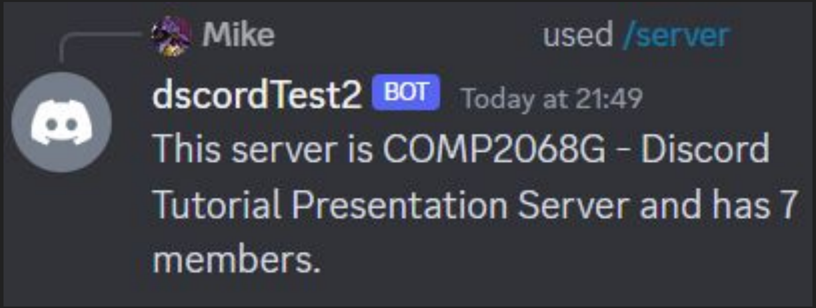


```
mikeb@Mikes-Linkwave-Laptop MINGW64 ~/OneDrive/Desktop/Lakehead Work/2022 Winter Se
$ node deploy-commands.js
Started refreshing 3 application (/) commands.
Successfully reloaded 3 application (/) commands.
```

- If it is successful and your index.js is still running in another terminal, your bot should now work! Go and test it with one of the new slash commands

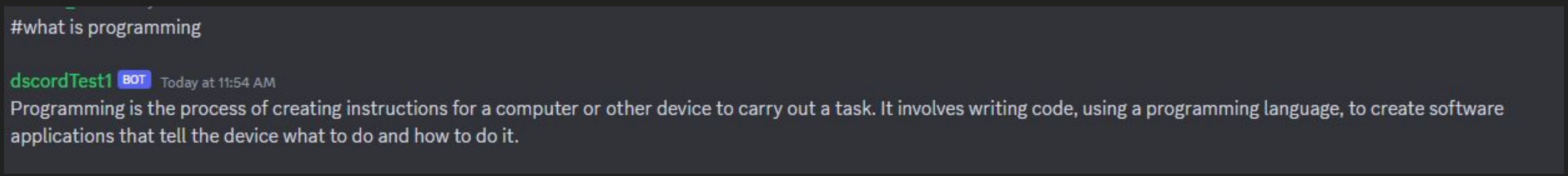


/server
Provides information about the server.
dscordTest2
/server



Mike used /server
dscordTest2 BOT Today at 21:49
This server is COMP2068G - Discord Tutorial Presentation Server and has 7 members.

Preview of our demo application!


#what is programming

dscordTest1 BOT Today at 11:54 AM
Programming is the process of creating instructions for a computer or other device to carry out a task. It involves writing code, using a programming language, to create software applications that tell the device what to do and how to do it.

- For our Demo Application we created a discord bot that uses ChatGPT to answer any question you ask the bot.
- To ask the bot a question you just use the prefix #

Our Bot’s Extra Features

- Command cooldowns
- Get input from the user
- ChatGPT Integration

Links and Resources

- Sample Exercise
- ChatGPT Bot Resource
- Developer Portal
- ChatGPT Site
- Creating Discord Bot with Node
- Discord.js npm package
- Discord.js Documentation
- Discord.js Public Server
- How to Create a Discord Bot on Developer Portal

Questions?