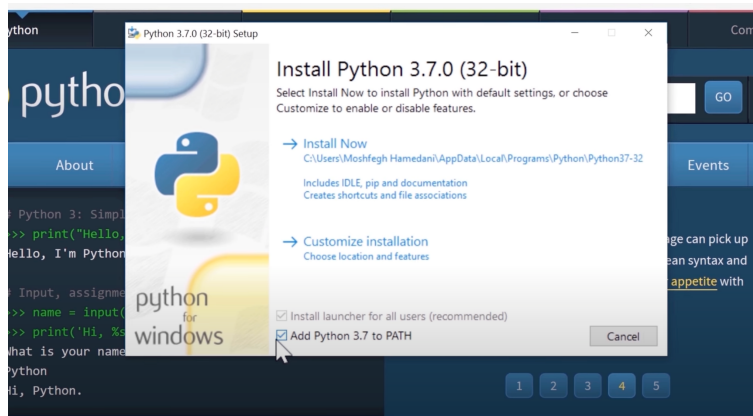


Setting Up Your Environment

Installing Python

1. Visit <https://www.python.org/>
2. MAC OS Users: Hit Downloads
3. Windows Users: make sure to select the checkbox titled "Add Python 3.7 to PATH" shown below:



Complete the rest of the installation steps once downloaded

Installing PyCharm

Download the Community version onto your laptop

<https://www.jetbrains.com/pycharm/download/#section=mac>

Pip Installations

Downloading Pip: run these commands in your terminal one after another

1. `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
2. `python3 get-pip.py`

Verifying You Have Pip

`pip3 --version` OR `pip --version`

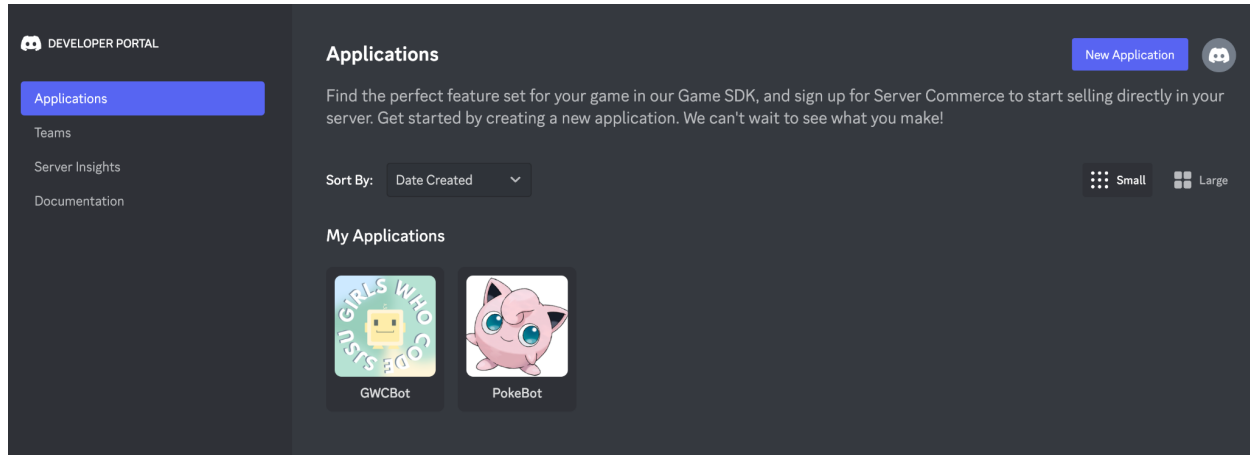
Install discord package

`pip install discord`

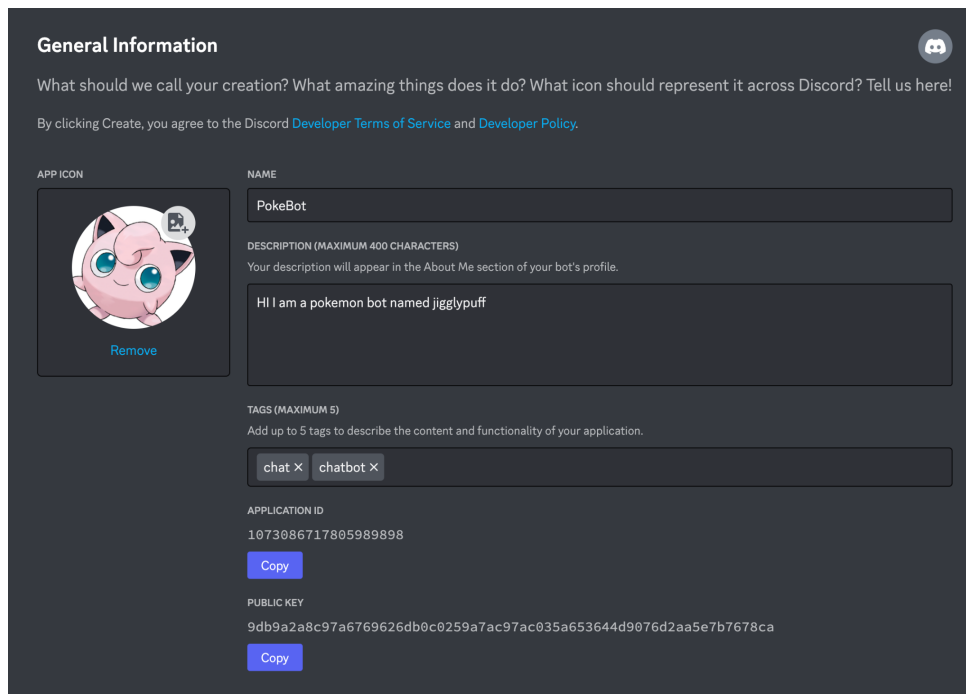
Configuring Discord Settings

****Make sure you are logged into your discord user on a web browser & have the discord app on your desktop****

1. Go to: <https://discord.com/developers/applications>
2. Click New Application



3. Give your bot general info
 - a. Name, Description
 - b. Tags: chat, chatbot



4. Navigate to OAuth2

The screenshot shows the Discord OAuth2 settings page. On the left sidebar, under 'SETTINGS', the 'OAuth2' option is selected. The main content area is titled 'OAuth2' and includes a description: 'Use Discord as an authorization system or use our API on behalf of your users. Add a redirect URI, pick your scopes, roll a D20 for good luck, and go!'. Below this is a link 'Learn more about OAuth2'. The 'Client information' section shows the 'CLIENT ID' as '1073086717805989898' with a 'Copy' button, and the 'CLIENT SECRET' with a 'Reset Secret' button. The 'Redirects' section states: 'You must specify at least one URI for authentication to work. If you pass a URI in an OAuth request, it must exactly match one of the URIs you enter here.' with an 'Add Redirect' button. The 'Default Authorization Link' section says: 'Pick the scopes and permissions your application needs to function, or add a custom URL. Other Discord users will be able to open your app's profile and directly add it to their server.' Below this is the 'AUTHORIZATION METHOD' dropdown set to 'In-app Authorization'. The 'SCOPES' section has two checkboxes: 'bot' (checked) and 'applications.commands' (unchecked).

5. Configure settings in OAuth2

- Authorization Method: In-app Authorization
- Scopes: bot
- Bot Permissions {check everything in the screenshot below}

This screenshot shows the 'Bot Permissions' section of the Discord OAuth2 settings. The 'AUTHORIZATION METHOD' is 'In-app Authorization'. The 'SCOPES' section has 'bot' checked and 'applications.commands' unchecked. The 'BOT PERMISSIONS' section is divided into three columns: 'GENERAL PERMISSIONS', 'TEXT PERMISSIONS', and 'VOICE PERMISSIONS'. In the 'GENERAL PERMISSIONS' column, 'Moderate Members' is checked, while all other permissions are unchecked. In the 'TEXT PERMISSIONS' column, all permissions are checked: 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, and Use Slash Commands. In the 'VOICE PERMISSIONS' column, all permissions are unchecked: Connect, Speak, Video, Mute Members, Deafen Members, Move Members, Use Voice Activity, Priority Speaker, Request To Speak, and Use Embedded Activities.

6. Navigate to URL Generator

- Repeat the process of selecting bot, and proper permissions
- COPY THE URL GENERATED AT THE BOTTOM OF THE SCREEN
- SAVE THE LINK SOMEWHERE SAFE!!! (Notes app, etc.)

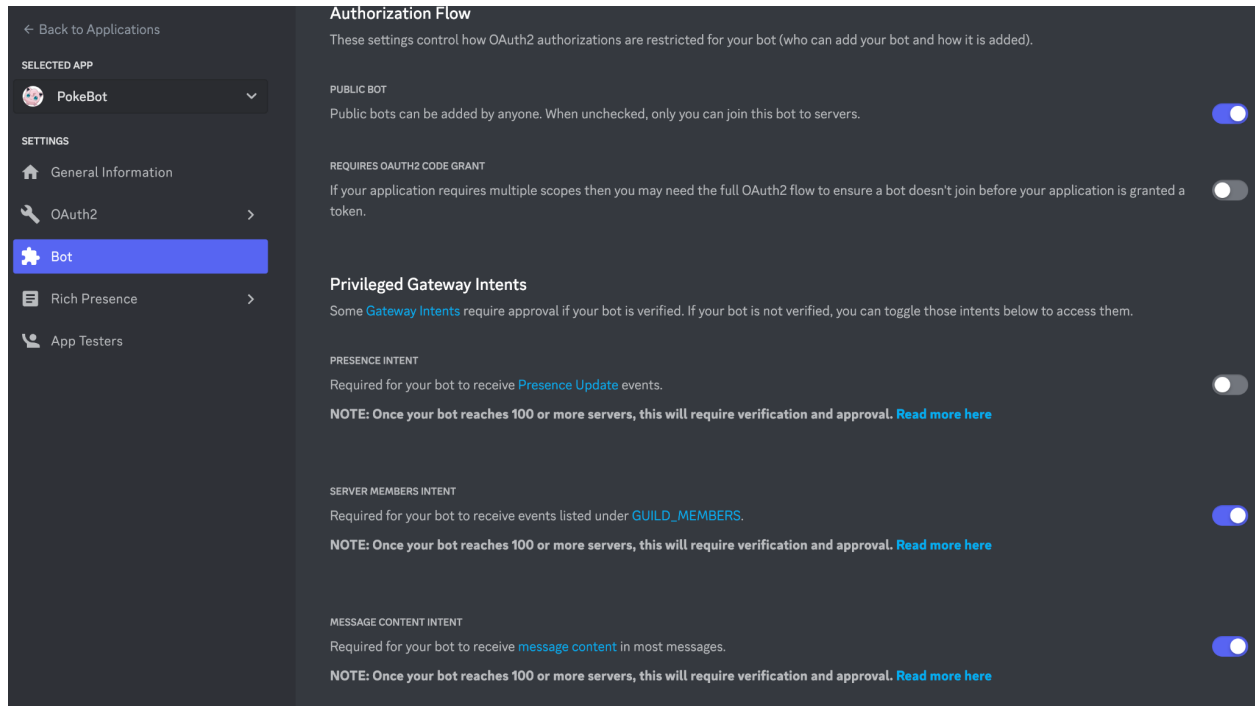
The screenshot shows the 'OAuth2 URL Generator' page. On the left sidebar, 'PokeBot' is selected under 'SELECTED APP', and 'OAuth2' is selected under 'SETTINGS'. The main area is titled 'OAuth2 URL Generator' and contains a list of permissions under 'SCOPES'. The 'bot' scope is checked. Below this, there are sections for 'BOT PERMISSIONS' (General, Text, Voice) and 'GENERATED URL'. The generated URL is: `https://discord.com/api/oauth2/authorize?client_id=107308677805989876&permissions=1632088041840scope=bot`. A 'Copy' button is next to the URL.

7. Navigate to Bot

- Generate your Token, **COPY your TOKEN & SAVE IT SOMEWHERE!**

The screenshot shows the 'Bot' settings page. On the left sidebar, 'PokeBot' is selected under 'SELECTED APP', and 'Bot' is selected under 'SETTINGS'. The main area is titled 'Bot' and contains a message: 'A new token was generated! Be sure to copy it as it will not be shown to you again.' Below this, there is a 'Build-A-Bot' section with fields for 'ICON' (a Jigglypuff image) and 'USERNAME' (PokeBot). The 'TOKEN' section shows a long alphanumeric string: `MTA3MzA4NjcNzgwNTk4OTg5OA.GBSzxp.OV_h6SjxqbkWAOB2JXlkjzTXeP-MfOOUCWPku`. There are 'Copy' and 'Reset Token' buttons. At the bottom, there is an 'Authorization Flow' section with a 'PUBLIC BOT' toggle switch.

ON THE SAME PAGE, WHEN YOU SCROLL, MAKE SURE ALL THESE SETTINGS ARE TURNED ON!! VERY VERY IMPORTANT FOR YOUR BOT TO PROPERLY WORK.



9. Paste the link your saved above onto a new browser
10. Authorize your bot to your discord, and any server you want it to (preferably an empty server you own)

Code!!

1. In PyCharm create a new project
2. Name your project: discordBot
3. Make a new python file: bot.py
4. In the terminal: pip install discord
 - a. This will give us all the files needed to create the bot

Bot.py

Step 1: create a function to run your discord bot

```
def run_discord_bot():
```

Step 2: Create a variable for your token, and paste your token key in this variable (the token that you saved in your previous steps)

```
TOKEN = 'insert your token key you generated'
```

Step 3: Intent settings that we enabled earlier configured with the client

```
intents = discord.Intents.default()
intents.message_content = True
client = discord.Client(intents=intents)
```

Step 4:

@client.event is a decorator for our function on_ready()

async def expression is the way to define custom coroutines:

- It stops the function's execution at that point and works on other things until it comes back to that point and finishes off its work
- This allows for your program to be doing multiple things at the same time without using threads or complicated multiprocessing

```
@client.event
async def on_ready():
    print(f'{client.user} is now running!')
```

Step 5:

Splicing string in python: [start:end]

`str(author)[:5]` // A discord username is always in format User#1234

await is another coroutine

```
@client.event
async def on_message(message):
    author = message.author
    author_name = str(author)[:5]
    if message.content.lower() == 'hello':
        await message.channel.send(f'Hello {author_name}')
```

Step 6: adding more if conditions / functionality to your bot

```
if message.content.lower() == 'roll':
    await message.channel.send(random.randint(1, 6))
```

Step 7: running your OATH Token client

```
client.run(TOKEN)
```

At the end, your bot.py file should look like this:

```
import discord
import random

def run_discord_bot():
    TOKEN = 'insert your token here '
    intents = discord.Intents.default()
    intents.message_content = True
    client = discord.Client(intents=intents)

    @client.event
    async def on_ready():
        print(f'{client.user} is now running!')

    @client.event
    async def on_message(message):
        author = message.author
```

```
author_name = str(author)[:5]

if message.content.lower() == 'hello':
    await message.channel.send(f'Hello {author_name}')
if message.content.lower() == 'roll':
    await message.channel.send(random.randint(1, 6))

client.run(TOKEN)
```


Main.py

Step 1: import your bot.py file

```
import bot
```

Step 2: define your main function, and run your bot function:

- `__main__` is the name of the environment where top-level code is run. “Top-level code” is the first user-specified Python module that starts running
 - It’s “top-level” because it **imports** all other modules that the program needs, such as **bot**. Sometimes “top-level code” is called an entry point to the application.

```
if __name__ == '__main__':  
    bot.run_discord_bot()
```

At the end your main.py file should look like this:

```
import bot  
  
if __name__ == '__main__':  
    bot.run_discord_bot()
```