



# UWB ACM Discord API Intro

UWB Hackathon  
25/May/2018

# BUT FIRST: Where Can I Find This?

Presentation too fast?

Need to Review?

Find this slidedeck here:

<https://bit.ly/2x9Ywx1>

( <https://uwb-acm.github.io/Hackathon-Docs/API-Presentation.html> )



# Presentation Overview

- I. What is an API?
- II. Provided Hackathon Content
- III. Bot Deployment Walkthrough



# What is an API?

- I. API?
- II. Library?
- III. How Can I Find Libraries?

## API GUIDE

REQUEST URL FORMAT:

`http://www.com/<username>/<item ID>`

SERVER WILL RETURN AN XML DOCUMENT WHICH CONTAINS:

- THE REQUESTED DATA
- DOCUMENTATION DESCRIBING HOW THE DATA IS ORGANIZED SPATIALLY

## API KEYS

TO OBTAIN API ACCESS, CONTACT THE X.509-AUTHENTICATED SERVER AND REQUEST AN ECDH-RSA TLS KEY...



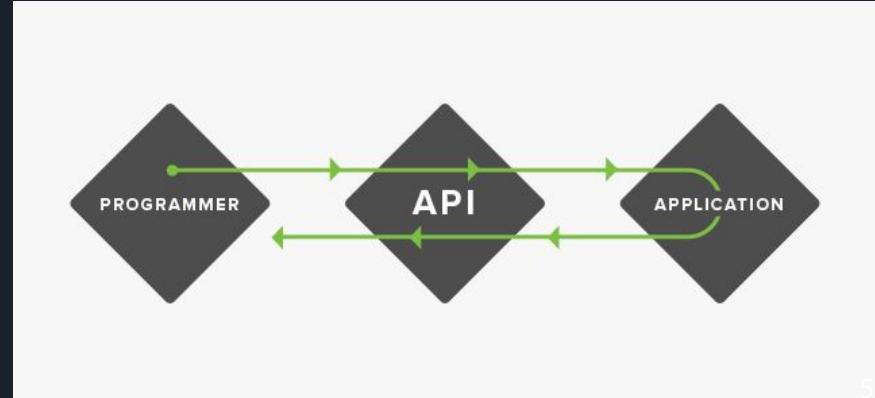
IF YOU DO THINGS RIGHT, IT CAN TAKE PEOPLE A WHILE TO REALIZE THAT YOUR "API DOCUMENTATION" IS JUST INSTRUCTIONS FOR HOW TO LOOK AT YOUR WEBSITE.

# I API?

“An API is an interface for other programs to interact with your program ... without having direct access.” - Anon

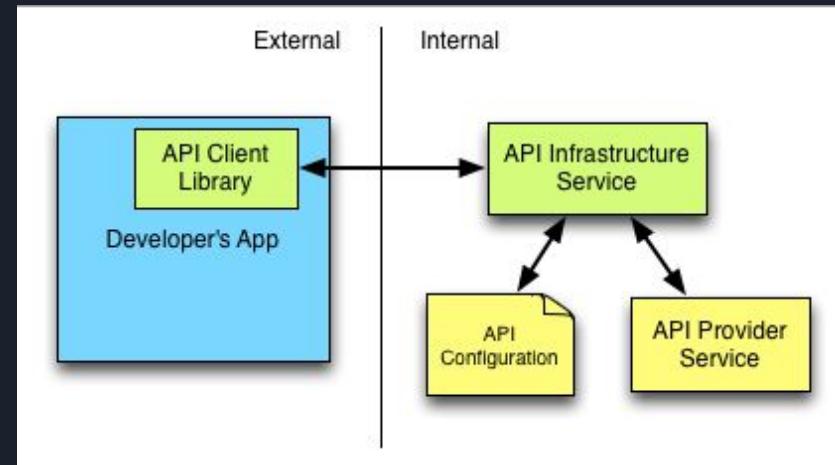
## Application Programming Interface

- Exposes interface for a service
- Connects decoupled services
- Lets us do cool stuff!



# Library?

Libraries make APIs more approachable!



*“Wouldn’t it be great if we could give them some code in their own language, to help them get started with the API?*

***That’s what a client library does.***

*It helps to reduce the amount of code ... ensures they’re using the API in the best supported manner.”* - Sarah Maddox

# How Can I Find Libraries?

Find library to suit your  
Language of choice!

## ***SEARCH GOOGLE***

“<API\_NAME> library”

StackOverflow

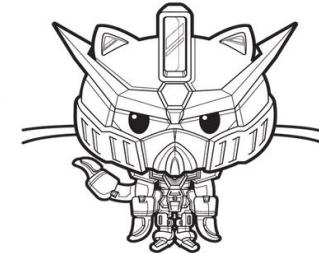
API Documentation

Usually contain links →

Octokit comes in  
many flavors

Use the official Octokit library, or choose between  
any of the available third party libraries.

Ruby [octokit.rb](#)  
.NET [octokit.net](#)  
Node.js [octokit/rest.js](#)



## Third-party libraries

Clojure [Tentacles](#)

Dart [github.dart](#)

Emacs Lisp [gh.el](#)

Erlang [octo.erl](#)

Go [go-github](#)

Haskell [github](#)

Octokit is a GitHub API

# Provided Hackathon Content

- I. Documentation
- II. Barebones Repositories
- III. Workshops
- IV. ???



# Documentation

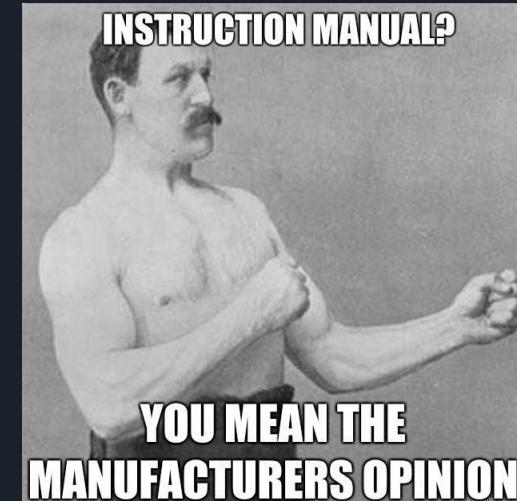
We have a whole bunch of stuff to help you get started!

<https://uwb-acm.github.io/Hackathon-Docs/>

Short-URL: <https://bit.ly/2IM1Gwx>

Including...

- Getting started with Discord Bot Development
- Useful APIs and Libraries
- Token Management
- Using Git Flow with Bot Development





# Barebones Repositories

- Resources to “bootstrap” into projects
- Fork and clone repository, add functionality
- Deploy bot

UWB ACM GitHub Profile

<https://bit.ly/2Lpq44q>

Discord Python Example Repository

<https://bit.ly/2IHPuwB>



# Workshops

Itinerary: <https://bit.ly/2LtS4Uv>

Gray Hats Workshop - 12:45pm

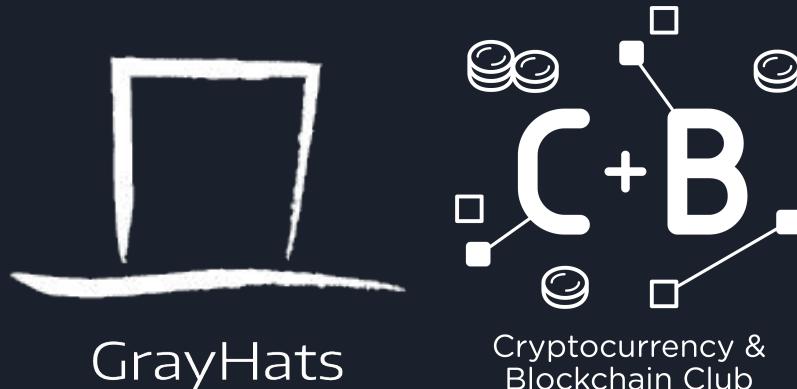
*Security, Bots, && Hacks*

"The Grayhats bot workshop will be two parts. The first part is a discord bot that focuses on reconnaissance of domains. The bot provides functionality to determine the domain owner, location, server info, and checks if its up. Some important concepts utilized by the bot are asynchronous IO for improved responsiveness, interaction with JSON apis, and utilizing external utilities in a safe manner. The second part is a slack bot used to remotely control a compromised computer. While having a focus on cybersecurity, both bots use concepts that can be applied anywhere."

Blockchain/Cryptocurrency Workshop - 2:30pm

*Blockchain, Ethereum, && Smart Contracts*

"The Cryptocurrency & Blockchain Club workshop will go over the basics of blockchain, Ethereum, and the Solidity programming language. Participants will learn what a blockchain is and why it's useful. They can then build a lottery smart contract that is interactive. The club looks forward to giving people an opportunity to learn about blockchain!"



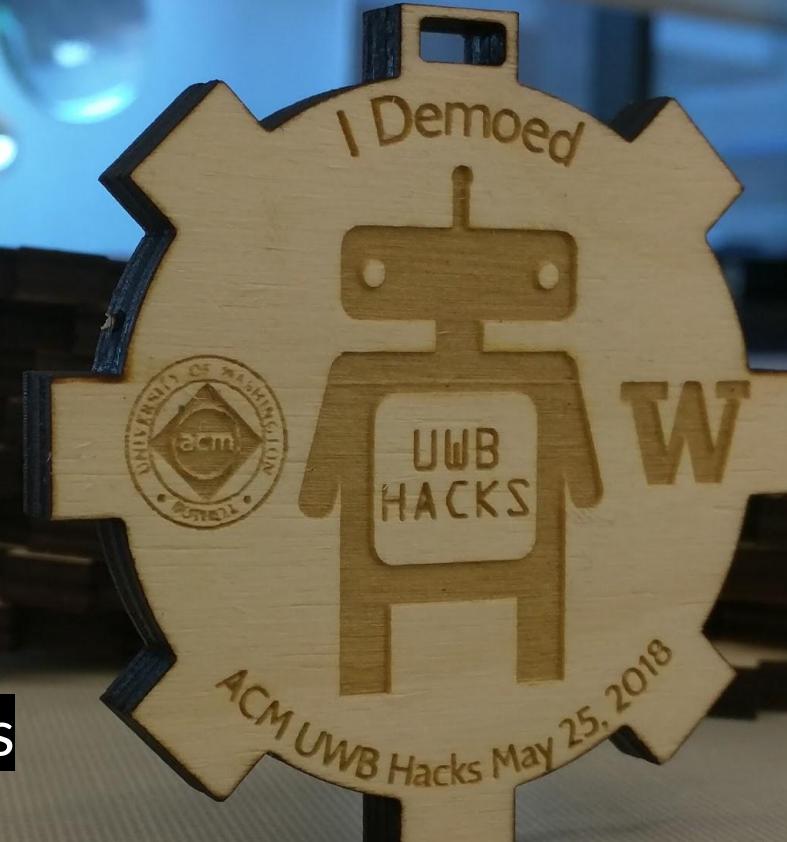
Cryptocurrency &  
Blockchain Club  
University of Washington Bothell

# Prizes

Prizes awarded during  
ending ceremony

Bot Tokens

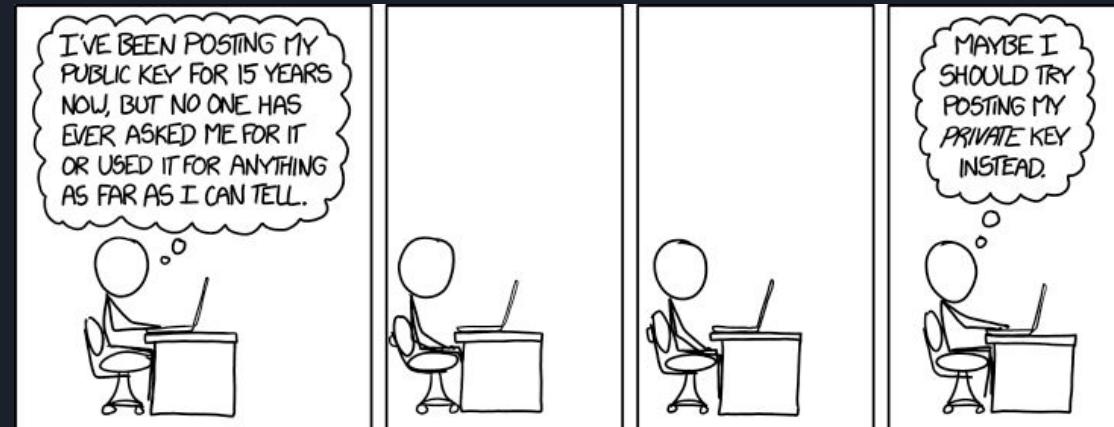
ACM Hosted repositories



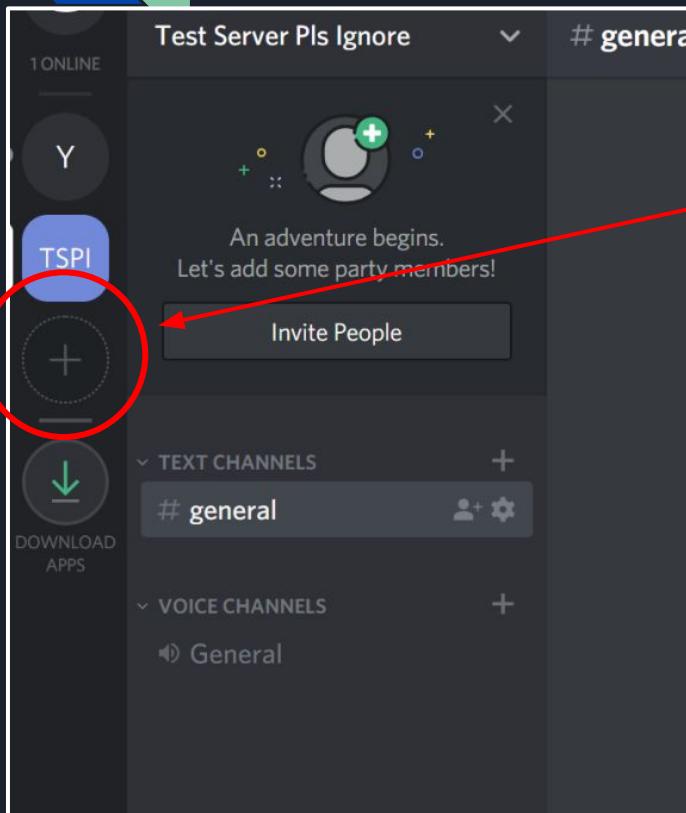
We add your project to our organization!  
You add it to your resume!

# Deployment Walkthrough

- I. Creating a Discord Server
- II. Generating an API key & Adding the Bot
- III. Forking and Cloning the Example Repository
- IV. Configuring a Bot
- V. Deploying a Bot



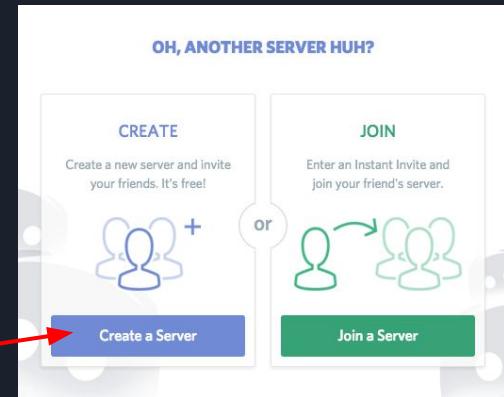
# Creating a Discord Server



<https://bit.ly/2gD5VKy>

Click this button

Select “Create a Server”



Fill out information

Make a fun name!

A screenshot of the "Create Your Server" form. It includes fields for "SERVER NAME" (with a placeholder line), "SERVER REGION" (set to "US West" with a "Change" button), and a circular "CHANGE ICON" button with a "Minimum Size: 128x128" note. At the bottom right is a "Create" button.

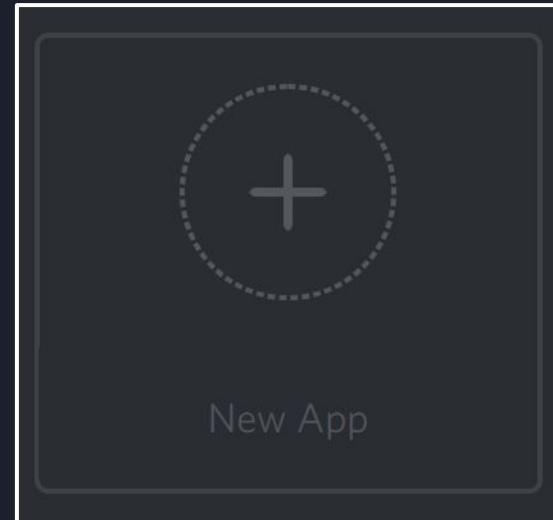


# Generating an API Key (i)

- Head over to the Discord Developer Documentation

<https://discordapp.com/developers/docs/intro>  
( <https://bit.ly/2nU8nQw> )

- Login!
- Navigate to `My Apps` page
- Click this button → → → →



# Generating an API Key (ii)

Fill out the `create app` window

Give it a fun name!

New App

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

APP NAME *	REDIRECT URI(S)
My Cool Bot!?	<a href="#">Add Redirect</a>
Something users will recognize and trust	
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 URLs you enter here. <a href="#">Learn more</a>	
APP DESCRIPTION	APP ICON
Oh yeah, this is a cool bot.	 Remove
Maximum 400 characters	
<a href="#">Cancel</a>	<a href="#">Create App</a>

# Generating an API Key (iii)

If success, then you'll see this →

REVIEWING: My Cool Bot!

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

**GREAT SUCCESS!**  
Your sweet new application has been created successfully!

APP DETAILS

Client ID: 441807202798534658  
Client Secret: [click to reveal](#)

APP NAME \*

My Cool Bot!?

Something users will recognize and trust

REDIRECT URI(S)

Add Redirect

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. [Learn more](#)

OAUTH2 URL GENERATOR

Generate OAuth2 URL

Need help calculating permissions for your bot, or generating an OAuth2 URL for your application? Click the button to use our handy tool.

Now, create a `Bot User` (below)

**Bot**

 You can bundle a Bot User with your app to interact with users in a more conversational manner. [This action is irreversible.](#) [Learn more about bot users.](#)

[Create a Bot User](#)

# Generating an API Key (iv)

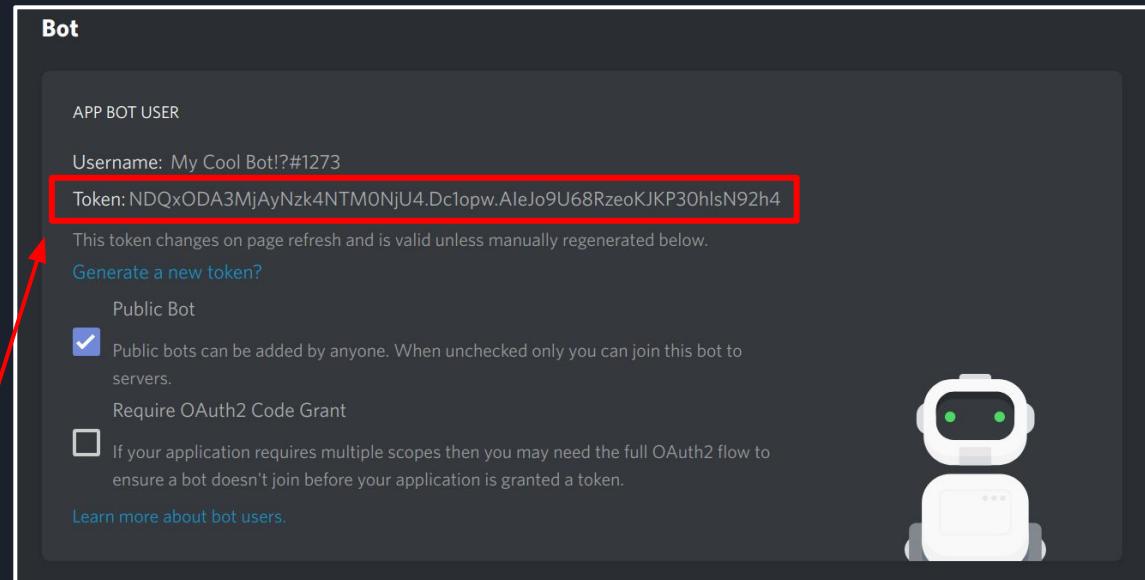
Now you have a bot!

This contains:

Bot's Client ID

Bot Token

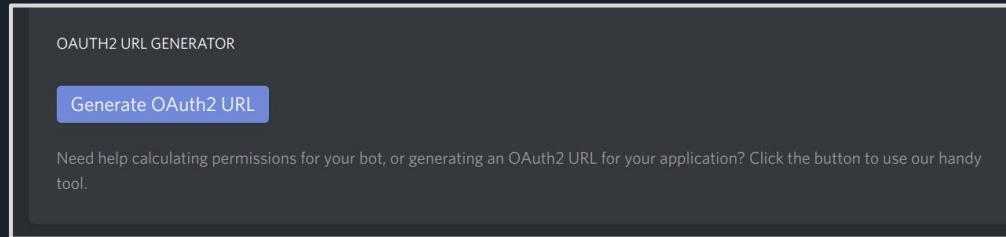
DON'T SHARE TOKEN (It's your private key!)





# Generating an API Key (v)

Generate the bot's invite link using the “Generate OAuth2 URL”



# Generating an API Key (vi)

Copy the link generated by this form. If we don't request permissions here, the bot will use the server defaults.

The screenshot shows the 'SCOPES' section of the Discord OAuth2 permission generation interface. It lists various permissions with checkboxes:

- identify
- email
- connections
- guilds
- guilds.join
- gdm.join
- rpc
- rpc.api
- rpc.notifications.read
- bot
- webhook.incoming
- messages.read

Below the scopes, a URL is displayed: [https://discordapp.com/api/oauth2/authorize?client\\_id=449461009447649291&permissions=8](https://discordapp.com/api/oauth2/authorize?client_id=449461009447649291&permissions=8). A blue 'COPY' button is next to the URL. The entire URL box is highlighted with a red border.

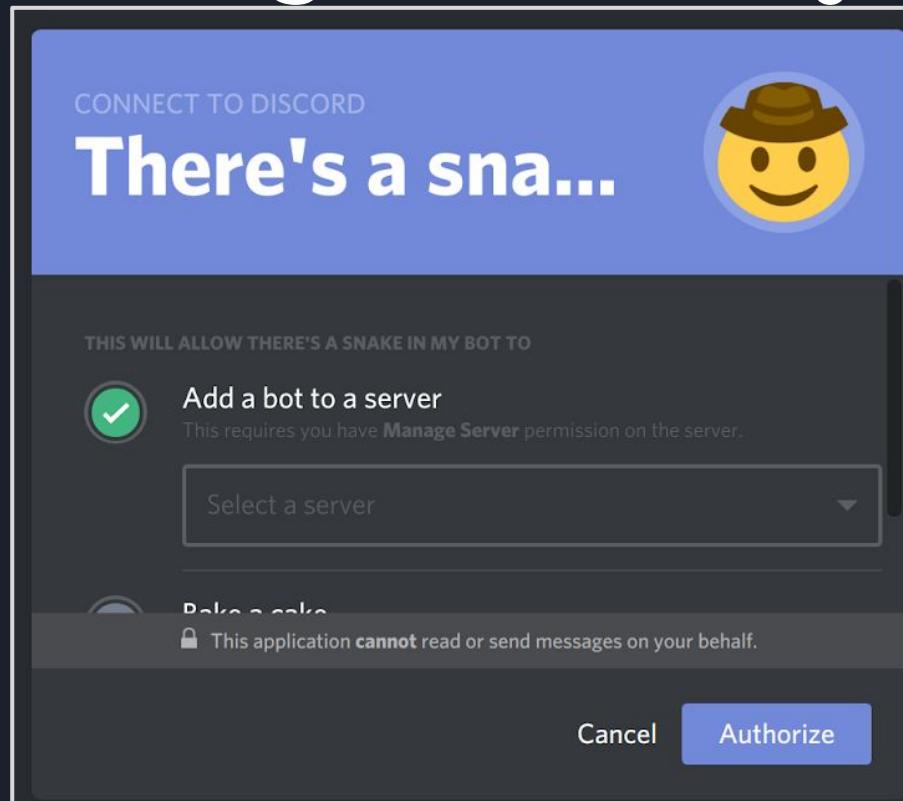
**Bot Permissions**

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

Open this in a new tab



# Generating an API Key (vii)



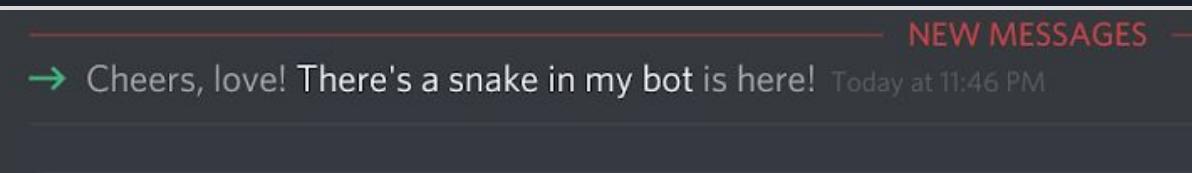
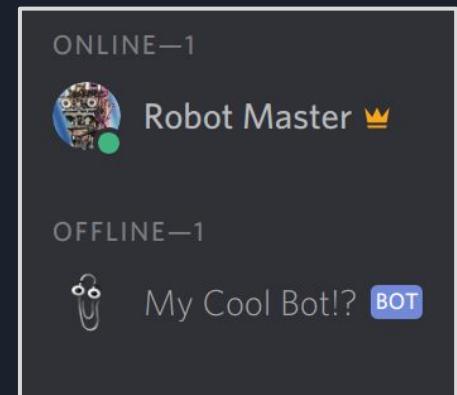
(It's supposed to say  
“There’s a snake in  
my bot”)

III

# Generating an API Key (viii)

Bot has joined the server!

By default, bots are offline



...now, let's activate our bot



# Forking and Cloning the Example Repository (Fork)

*Time to use ACM's barebones repositories*

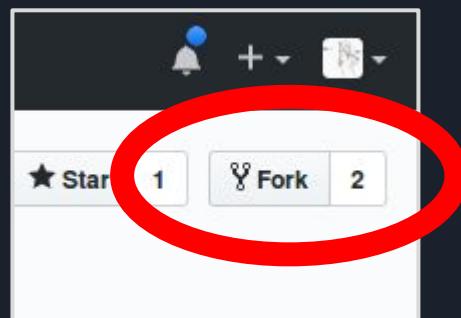
Login to GitHub

Fork Repository

Go to ACM Discord-Python-Example Repo

<https://bit.ly/2IHPuwB>

Press Fork button →→→→→→→→→→



THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.





# Forking and Cloning the Example Repository (Clone)

*Time to use ACM's barebones repositories*

Navigate to directory where you want to clone repo in

```
cd ~/<path_to_dir>
```

Clone repo from git (use your USERNAME in URL)

```
git clone https://github.com/USERNAME/Discord-Python-Example.git
```

Navigate into repository

```
cd Discord-Python-Example
```

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



# Forking and Cloning the Example Repository (Clone)

You should clone from your own fork, not this one



```
chris@impostor ~/git $ git clone https://github.com/UWB-ACM/Discord-Python-Example
Cloning into 'Discord-Python-Example'...
remote: Counting objects: 289, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 289 (delta 4), reused 10 (delta 4), pack-reused 277
Receiving objects: 100% (289/289), 57.15 KiB | 0 bytes/s, done.
Resolving deltas: 100% (160/160), done.
Checking connectivity... done.
chris@impostor ~/git $ ls
Discord-Python-Example
chris@impostor ~/git $ cd Discord-Python-Example/
chris@impostor ~/git/Discord-Python-Example $ ls
cogs LICENSE.md main.py README.md requirements.txt
chris@impostor ~/git/Discord-Python-Example $ █
```

# Configuring a Bot (i)

## Setup dependencies

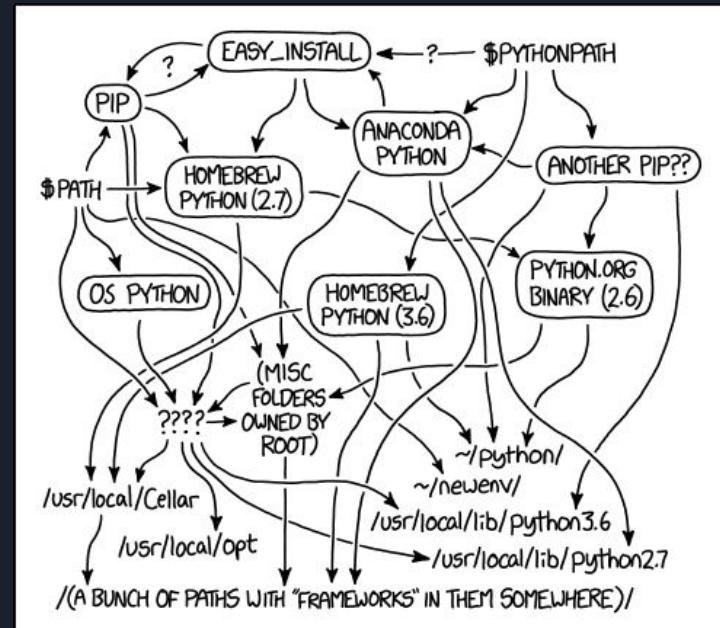
From project directory:

```
python3 -m pip install -r requirements.txt
```

Or, for you Windows users:

```
py -3 -m pip install -r requirements.txt
```

Be sure to run as root/administrator (use sudo)



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED  
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

# Configuring a Bot (ii)

Installing without root == angry red text

```
chris@impostor ~/git/Discord-Python-Example
File Edit View Search Terminal Help
running egg_info
creating discord.py.egg-info
writing top-level names to discord.py.egg-info/top_level.txt
writing requirements to discord.py.egg-info/requirements.txt
writing dependency_links to discord.py.egg-info/dependency_links.txt
writing discord.py.egg-info/PKG-INFO
writing manifest file 'discord.py.egg-info/SOURCES.txt'
reading manifest file 'discord.py.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
warning: no files found matching 'README.md'
writing manifest file 'discord.py.egg-info/SOURCES.txt'
creating build/lib/discord/bin
copying discord/bin/libopus-0.x64.dll -> build/lib/discord/bin
copying discord/bin/libopus-0.x86.dll -> build/lib/discord/bin
running install_lib
creating /usr/local/lib/python3.5/dist-packages/discord
error: could not create '/usr/local/lib/python3.5/dist-packages/discord': Permission denied

-----
Command "/usr/bin/python3 -u -c "import setuptools, tokenize; __file__ ='/tmp/pip-install-ilebgx7d/discord.py/setup.py';f=getattr(tokenize, 'open', open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code, __file__, 'exec'))" install --record /tmp/pip-record-s0pq3vtq/install-record.txt --single-version-externally-managed --compile" failed with error code 1 in /tmp/pip-install-ilebgx7d/discord.py
chris@impostor ~/git/Discord-Python-Example $ 
```

# Configuring a Bot (iii)

Installing with root works, will take a few seconds to install.

```
chris@impostor ~/git/Discord-Python-Example
File Edit View Search Terminal Help
elements.txt (line 1)) (3.4)
requirement already satisfied: PyNaCl==1.1.2 in /usr/local/lib/python3.5/dist-packages (from discord.py[voice]->-r requirements
txt (line 1)) (1.1.2)
requirement already satisfied: async-timeout>=1.2.0 in /usr/local/lib/python3.5/dist-packages (from aiohttp<2.3.0,>=2.0.0->disc
ord.py[voice]->-r requirements.txt (line 1)) (2.0.1)
requirement already satisfied: chardet in /usr/local/lib/python3.5/dist-packages (from aiohttp<2.3.0,>=2.0.0->discord.py[voice]
->-r requirements.txt (line 1)) (3.0.4)
requirement already satisfied: multidict>=2.1.4 in /usr/local/lib/python3.5/dist-packages (from aiohttp<2.3.0,>=2.0.0->discord.
py[voice]->-r requirements.txt (line 1)) (4.1.0)
requirement already satisfied: yarl>=0.11 in /usr/local/lib/python3.5/dist-packages (from aiohttp<2.3.0,>=2.0.0->discord.py[voi
ce]->-r requirements.txt (line 1)) (1.1.1)
requirement already satisfied: cffi>=1.4.1 in /usr/local/lib/python3.5/dist-packages (from PyNaCl==1.1.2->discord.py[voice]->-r
requirements.txt (line 1)) (1.11.5)
requirement already satisfied: six in /usr/local/lib/python3.5/dist-packages (from PyNaCl==1.1.2->discord.py[voice]->-r require
ments.txt (line 1)) (1.11.0)
requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.5/dist-packages (from yarl>=0.11->aiohttp<2.3.0,>=2.0.0->dis
cord.py[voice]->-r requirements.txt (line 1)) (2.6)
requirement already satisfied: pycparser in /usr/local/lib/python3.5/dist-packages (from cffi>=1.4.1->PyNaCl==1.1.2->discord.py
[voice]->-r requirements.txt (line 1)) (2.18)
installing collected packages: discord.py
  Running setup.py install for discord.py ... done
successfully installed discord.py-1.0.0a1467+g6b75179
chris@impostor ~/git/Discord-Python-Example $
chris@impostor ~/git/Discord-Python-Example $ sudo python3 -m pip install -r requirements.txt
```



# Configuring a Bot (iv)

## Setup Bot's API token

- Create a new file in the same directory as the bot called config.ini
- Copy Bot Token to clipboard (Ctrl-C)
- Put following text into config.ini file:

```
[Configuration]
```

```
connection_token=PUT_YOUR_BOT_TOKEN_HERE
```

!!! Be sure config.ini is in your .gitignore file !!! (Don't upload your private key!)

# Configuring a Bot (v)

The image shows two side-by-side screenshots. On the left is the Discord developer portal's 'Bot' configuration page. It displays a token field containing a long string of characters, which is highlighted with a green box. An orange arrow points from this highlighted token to the right screenshot, indicating where it should be pasted. On the right is a code editor showing a configuration file named 'config.ini'. The file contains a section '[Configuration]' with a single key-value pair: 'connection\_token=NDQ5NDYxMDA5NDQ3NjQ5Mjkx.DelA3w.PZmU6-3AxSji34pTs\_DJxGo0c5A'. The bottom status bar of the code editor shows '.ini' and 'Ln 2, Col 77'.

Maximum 400 characters

Remove

**Bot**

APP BOT USER

Username: There's a snake in my bot#8730

Token:

```
NDQ5NDYxMDA5NDQ3NjQ5Mjkx.DelA3w.PZmU6-3AxSji34pTs_DJxGo0c5A
```

This token changes on page refresh and is valid unless manually regenerated below.

[Generate a new token?](#)

**Public Bot**

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

**Require 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.

[Learn more about bot users.](#)

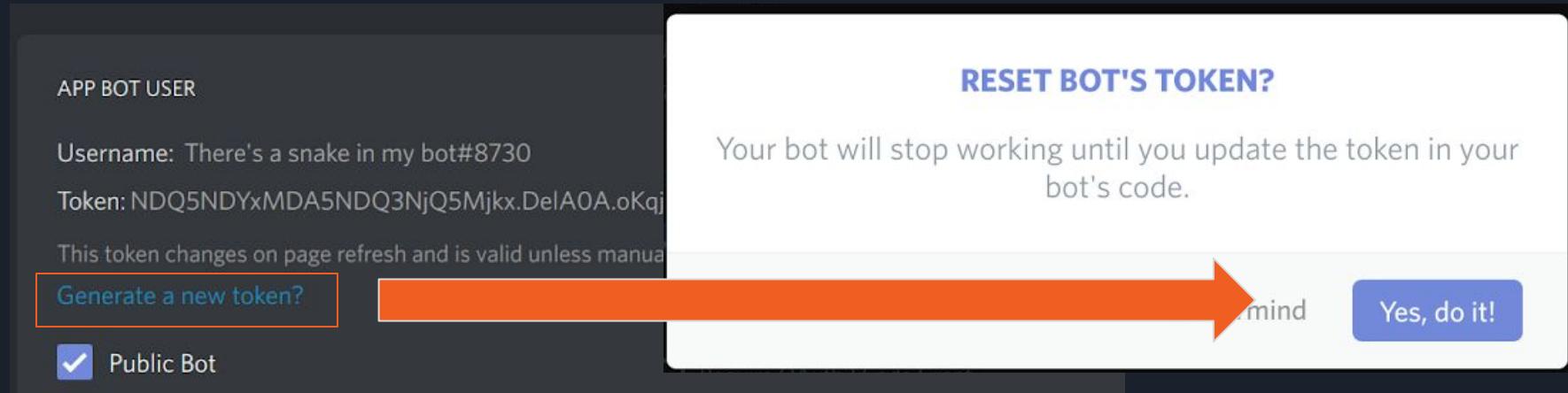
File Edit View Search Tools Documents Help

[Configuration]  
connection\_token=NDQ5NDYxMDA5NDQ3NjQ5Mjkx.DelA3w.PZmU6-3AxSji34pTs\_DJxGo0c5A

.ini ▾ Tab Width: 4 ▾ Ln 2, Col 77 IN

# Configuring a Bot (vi)

By the way, if you leak your token like we just did





# Deploying a Bot

From project directory:

```
python3 main.py
```

That was hard...



# Deploying a Bot

```
chris@impostor ~/git/Discord-Python-Example $ ls
cogs config.ini LICENSE.md main.py README.md requirements.txt
chris@impostor ~/git/Discord-Python-Example $ python3 main.py
using discordpy version 1.0.0a
Logged in There's a snake in my bot - 449461009447649291
Version 1.0.0a
□
```

→ Cheers, love! There's a snake in my bot is here! Today at 11:46 PM

11:47 PM You! >>help

11:47 PM BOT There's a snake in my bot

[https://github.com/UWB-ACM/CSSBot\\_Py](https://github.com/UWB-ACM/CSSBot_Py)

BasicCog:  
ping  
No Category:  
help Shows this message.

Type >>help command for more info on a command.  
You can also type >>help category for more info on a category.

11:47 PM You! >>ping

11:47 PM BOT There's a snake in my bot Pong!





# How Do I add functions? (i)

What are cogs?

Cogs are plugins for your bot!

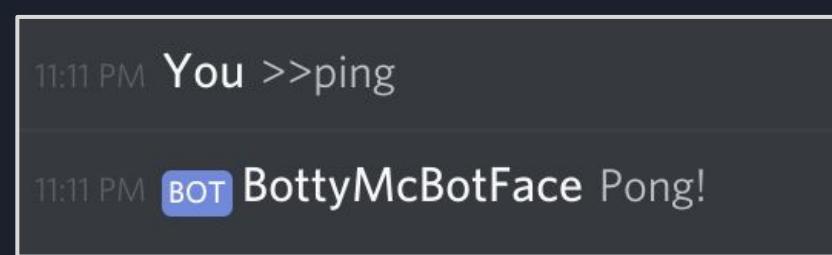
What is a command prefix?

Short string to signal bot

Be unique!

Else, bots could have conflicts

e.g., ">>"



# How Do I add functions? (ii)

Where are cogs declared?

main.py

<https://bit.ly/2kph7vl>

Where are cogs defined?

Discord-Python-Example/cogs/<COG\_NAME>.py





# How Do I add functions? (iii)

Discord-Python-Example/cogs/basic.py

```
9 # setup
10 class BasicCog:
11     def __init__(self, bot):
12         self.bot = bot
13
```

Discord-Python-Example/main.py

```
36
37 # this is where extensions are added by default
38 default_extensions = ['cogs.basic']
39
```

<https://bit.ly/2xbbqUb>

```
16 @commands.command()
17 async def ping(self, ctx):
18     await ctx.send("Pong!")
19
```

```
33
34 # add this cog to the bot
35 def setup(bot):
36     bot.add_cog(BasicCog(bot))
37
```

<https://bit.ly/2s8L1IU>



# How Do I add functions? (iv)

## Adding cogs:

- Name your cog
  - Here we use `my_awesome_cog`
- Add name of cog to list in `main.py`
  - `default_extensions = ['cogs.basic', 'cogs.my_awesome_cog']`
- Create `Discord-Python-Example/cogs/my_awesome_cog.py`
- Add commands to it!

# How Do I add functions? (v)

One of the types of commands you can make:

```
# command to wave hello
@commands.command()
async def wave(self, context):
    """
    replies back to the user who invoked the command with
    hello @user!
    """
    await context.send("Hello " + context.author.mention + "!")
```

You! >>wave

**BOT** There's a snake in my bot Hello @You!!



# How Do I add functions? (vi)

## SUMMARY

Cog files listed on this line:

<https://github.com/UWB-ACM/Discord-Python-Example/blob/barebones/main.py#L38>

Add functions to specified files:

<https://github.com/UWB-ACM/Discord-Python-Example/blob/barebones/cogs/basic.py#L30>

## Further reading:

Appendix - <https://uwb-acm.github.io/Hackathon-Docs/Getting-Started-With-Discord-Bot-Development-A-Walkthrough.html>

API Docs - <http://discordpy.readthedocs.io/en/latest/faq.html#commands-extension>

The repo - <https://github.com/UWB-ACM/Discord-Python-Example>



QUESTIONS?  
COMMENTS?  
CONCERNS?



# UWB Hacks

## TIMELINE

9:15 AM

Registration & Morning Snack (Panera)

10:00 AM

Opening Ceremony / Bot Introductions

12:00 PM

Lunch (Dominos)

12:45 PM

GrayHats Workshop

2:30 PM

Block Chain Workshop

6:00 PM

Dinner (Chipotle) / Mental Break

9:30 PM

Project Presentations

10:30 PM

Closing Ceremony