

UWB ACM

Discord API Intro

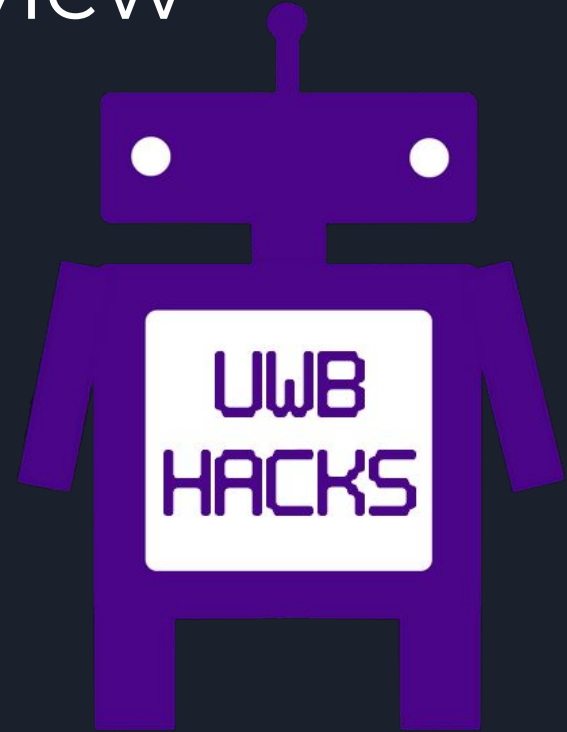


UWB Hackathon
25/May/2018



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...



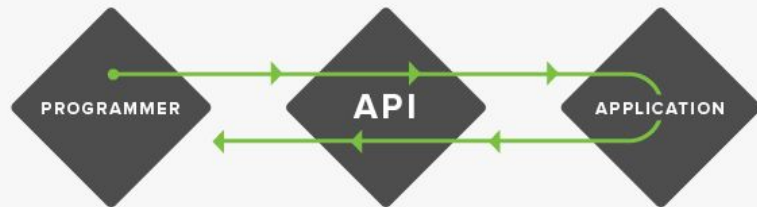
IF YOU DO THINGS RIGHT, IT CAN TAKE
PEOPLE A WHILE TO REALIZE THAT YOUR
"API DOCUMENTATION" IS JUST INSTRUCTIONS
FOR HOW TO LOOK AT YOUR WEBSITE.

API?

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

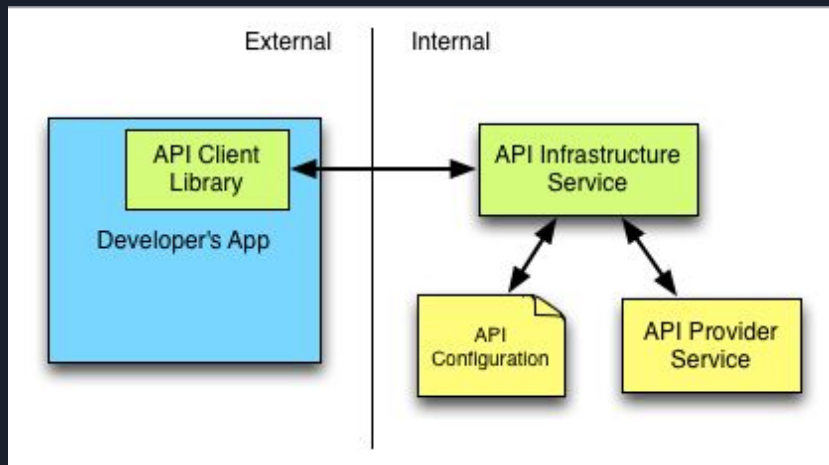
Application Programming Interface

- 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?

GOOGLE

“<API_NAME> library”

StackOverflow

API Documentation

Usually contain links →

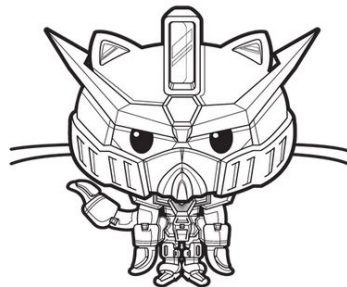
Octokit comes in many flavors

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](#)

Provided Hackathon Content

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



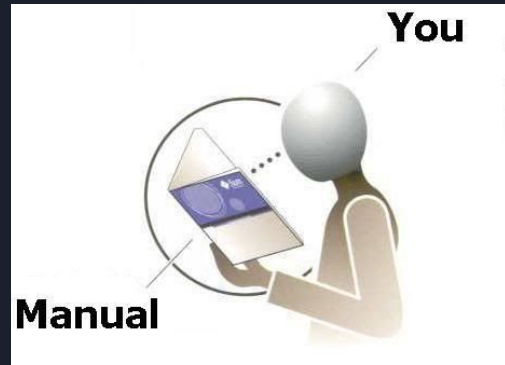
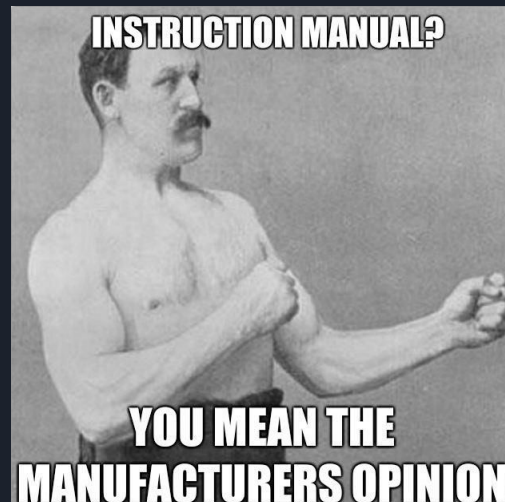
Documentation

We have a whole bunch of stuff!

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

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
- Clone repository, add functionality
- Deploy bot

<https://github.com/UWB-ACM>

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





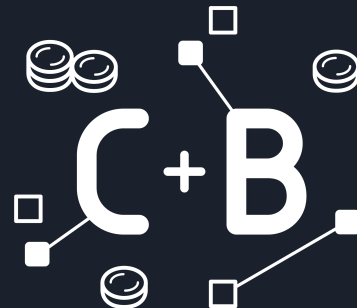
Workshops

Gray Hats Workshop - 12:45pm

Security, Bots, && Hacks



GrayHats.io



Cryptocurrency &
Blockchain Club
University of Washington Bothell

“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!”

Prizes

**Prizes awarded during
ending ceremony**

Bot Tokens

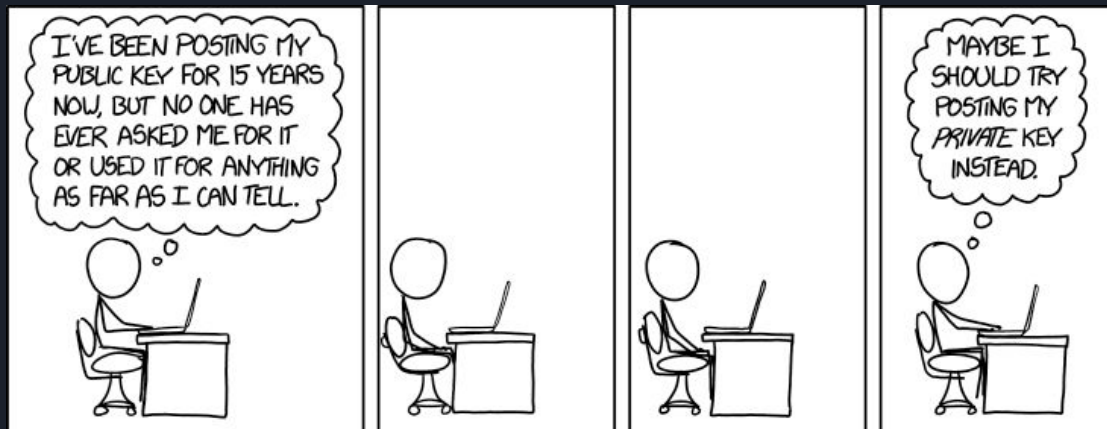
ACM Hosted repositories

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



Deployment Walkthrough

- I. Cloning a Repository
- II. Generating an API key
- III. Configuring a Bot
- IV. Creating a Discord Server
- V. Deploying a Bot





BUT FIRST:

Where Can I Find This?

Presentation too fast?

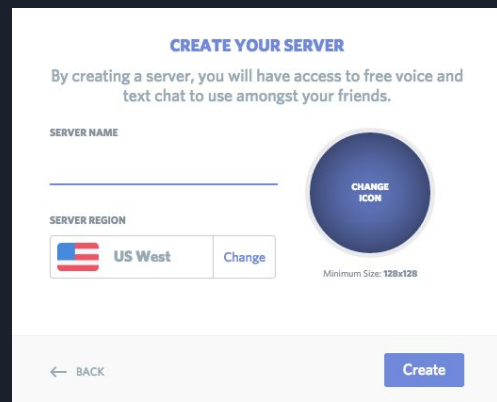
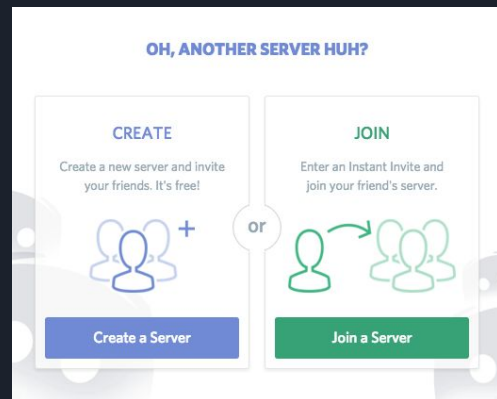
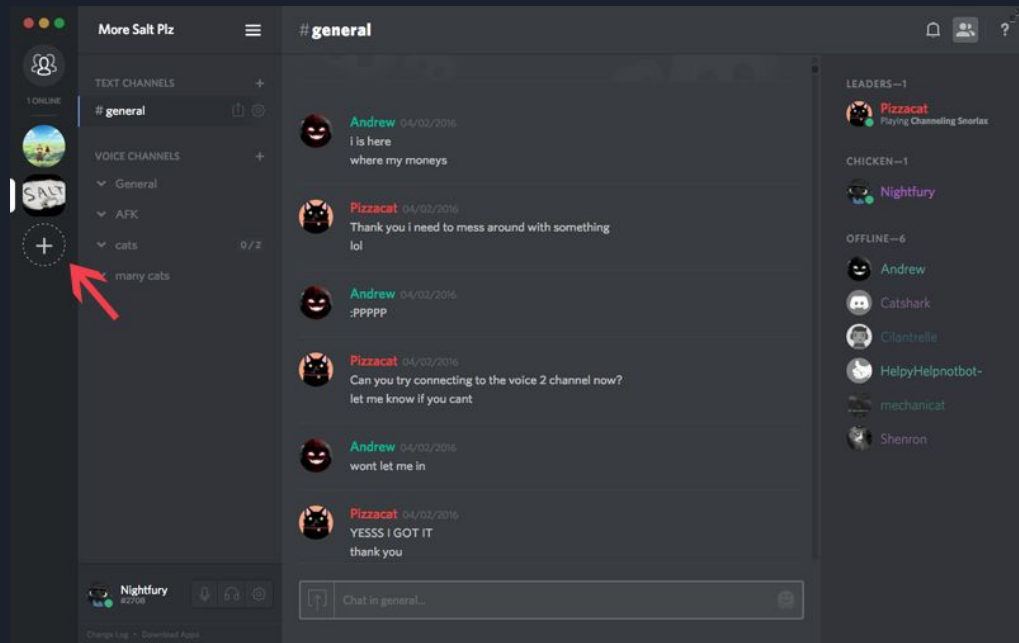
Need to Review?

Find this slidedeck here:

<URL>

Creating a Discord Server

<https://support.discordapp.com/hc/en-us/articles/204849977-How-do-I-create-a-server->

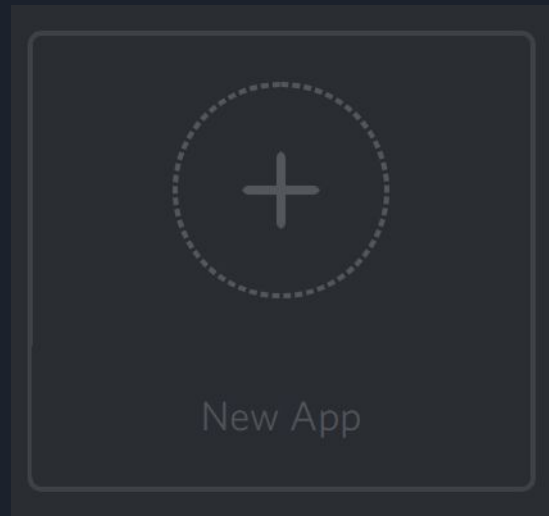


Generating an API Key (i)

- First, head over to the Discord Developer Documentation

<https://discordapp.com/developers/docs/intro>

- 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](#)

<p>APP NAME *</p> <p>My Cool Bot!?</p> <p>Something users will recognize and trust</p>	<p>REDIRECT URI(S)</p> <p>Add Redirect</p> <p>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</p>
<p>APP DESCRIPTION</p> <p>Oh yeah, this is a cool bot.</p> <p>Maximum 400 characters</p>	<p>APP ICON</p>  <p>Remove</p>

Cancel Create App

Generating an API Key (iii)

If success, then you'll see this →

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

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.

Generating an API Key (iv)

Now you have a bot!

This contains:

Bot's Client ID

Bot Token

Bot

APP BOT USER

Username: My Cool Bot!#1273

Token: NDQxODA3MjAyNzk4NTMONjU4.Dc1opw.AleJo9U68RzeoKJKP30hlsN92h4

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.](#)



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

Generating an API Key (v)

Time to add our bot to the server!

- Generate OAuth2 URL

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.

- Copy/Paste generated URL into browser
 - Add to the server we created

Bot is offline, by default. Now, let's activate!

ONLINE—1



Robot Master 🏆

OFFLINE—1



My Cool Bot!? BOT

Cloning a Repository

Time to use ACM's barebones repositories

Navigate to target directory

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

Clone repo from git

```
git clone https://github.com/UWB-ACM/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 MEMORIZE 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.



Configuring a Bot (i)

Setup dependencies

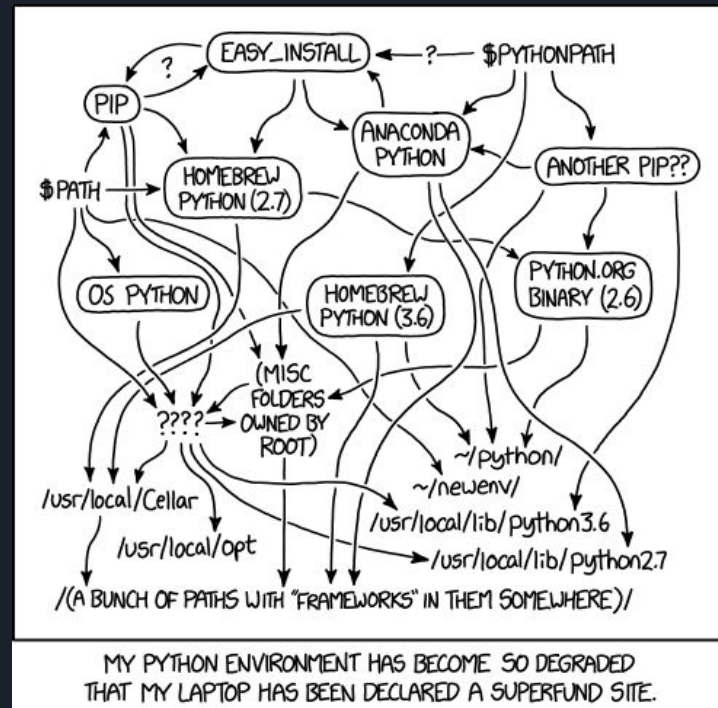
From project directory:

```
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`)





Configuring a Bot (ii)

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!)



Deploying a Bot

From project directory:

```
python3 main.py
```



How Do I add functions? (i)

What are `cogs`?

Cogs are essentially plugins for your bot!

What is a command prefix?

Short string to signal bot

e.g., ">>"

Me: >>ping

Bot: Pong!



How Do I add functions? (ii)

Where are cogs declared?

`main.py`

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

Where are cogs defined?

`Discord-Python-Example/cogs/<COG_NAME>.py`

How Do I add functions? (iii)

Discord-Python-Example/main.py

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

Discord-Python-Example/cogs/basic.py

```
9 # setup
10 class BasicCog:
11     def __init__(self, bot):
12         self.bot = bot
13
14     # ping command
15     @commands.cooldown(1, 30, commands.BucketType.user)
16     @commands.command()
17     async def ping(self, ctx):
18         await ctx.send("Pong!")
19
20     # echo command
21     # commands are given a name implicitly, but you can also define this explicitly as well
22     # the cooldown attribute allows you to limit how frequently the command can be used in
23     # a channel, a server, or by a specific user
24     @commands.cooldown(1, 60, commands.BucketType.user)
25     @commands.command(name='echo')
26     async def echo(self, ctx, *, message: str='_echo_'):
27         await ctx.send(ctx.author.mention + ' : ' + message)
28
29     # sqrt command
30     @commands.command(name='sqrt')
31     async def sqrt(self, ctx, *, num: int=0):
32         await ctx.send(_sqrt(num))
33
34 # add this cog to the bot
35 def setup(bot):
36     bot.add_cog(BasicCog(bot))
37
38 def _sqrt(num: float):
39     """
40     Example of how to include methods that can be easily tested
41
42     >>> _sqrt(4)
43     'Sqrt(4) = 2.0'
44
45     """
46
47     return 'Sqrt(' + str(num) + ') = ' + str(math.sqrt(num))
48
```

How Do I add functions? (iv)

Adding cogs:

- 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!

A command Thomas made
to generate truth tables →

```
31 @commands.cooldown(3, 30, commands.BucketType.user)
32 @commands.command()
33 async def bool(self, ctx, *, exp):
34     '''Creates truth table from expression using !,*,^,+
35
36     Wrap expression in "quotes" to include spaces in your expression.
37     If there are spaces in your expression and it is not properly wrapped,
38     then it will not process correctly, if at all.
39
40     Single chars are variables, more than one char in a row is invalid.
41     More than one ! in a row is invalid... fight me.
42     More than 5 variables is invalid, otherwise the messages would be long.
43     '''
44     await ctx.send(get_bool_table(exp))
45
46 def setup(bot):
47     '''Sets up the cog'''
48     bot.add_cog(HardwareUtilsCog(bot))
49
```



How Do I add functions? (v)

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