

Forum:[Hack Forums](#)

Thread: [Discord bot assistance needed.](#)

Mr. Robot - March 5, 2023, 6:03 pm

Having some issues.

Long story short, I have a feed.txt file I want to read that looks something like this

Code IT Security News Daily Summary 2023-03-05

<https://www.itsecuritynews.info/it-security-news-daily-summary-2023-03-05/>

Week in review: LastPass breach, GCP data exfiltration, UEFI bootkit & middot; Cyberattack on British Retailer WH Smith Exposes Employees` Data.

Week in review: LastPass breach, GCP data exfiltration, UEFI bootkit - Help Net Security

<https://www.helpnetsecurity.com/2023/03/05/week-in-review-lastpass-breach-gcp-data-exfiltration-uefi-bootkit/>

Here's an overview of some of last week's most interesting news, articles, interviews and videos: Google Cloud Platform allows data exfiltration ...

Sodium-Ion Battery for Stationary Energy Storage Market Regional Overview 2023-2028

<https://www.digitaljournal.com/pr/news/sodium-ion-battery-for-stationary-energy-storage-market-regional-overview-2023-2028-size-and-share-estimation-revenue-business-prospect-growth-opportunity-challenges-potential-benefits-till-2028>

It precisely gives the necessary data and its state of the art investigation to help figure out the ideal business ... Data Exfiltration Market.

Global Population Health Management Market Growth Statistics 2023 - Digital Journal

<https://www.digitaljournal.com/pr/news/global-population-health-management-market-growth-statistics-2023-competitive-landscape-restraining-factors-market-concentration-rate-development-status-and-growth-by-forecast-2027>

The goal of the PHM solution is to enable data-driven healthcare at the ... Global Data Exfiltration Market analysis by Top Leading Player ...

I want each article to be posted to discord as a separate message when the user types in !post_feed

I couldn't get it working so I made a split.py file to split each article into individual files 1.txt, 2.txt, 3.txt so forth in a folder called articles. Then was trying to send each one of those as an individual message. Still no luck.

The best I got was ['Found file: 1.txt', 'Found file: 2.txt']

If anyone knows a simpler way of doing this, that would be fucking fantastic.

My current setup is:

#The only thing you should need to edit is bot.py TOKEN = 'TOKENGOESHERE'

main.py

Code import bot

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

bot.py

Code import discord

import responses

```
async def send_message(message, user_message, is_private):  
    try:  
        response = responses.get_response(user_message)  
        await message.author.send(response) if is_private else await  
message.channel.send(response)  
  
    except Exception as e:  
        print(e)
```

```
def run_discord_bot():  
    TOKEN = 'TOKENGOESHERE'  
    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):  
    if message.author == client.user:  
        return
```

```
    username = str(message.author)  
    user_message = str(message.content)  
    channel = str(message.channel)
```

```
    print(f'{username} said: "{user_message}" ({channel})')
```

```
    if user_message[0] == '?':  
        user_message = user_message[1:]  
        await send_message(message, user_message, is_private=True)  
    else:  
        await send_message(message, user_message, is_private=False)
```

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

responses.py

Code import os

import random

from typing import List

```
def get_response(message: str) -> List[str]:
    p_message = message.lower()

    if p_message == '!post_feed':
        articles_dir = './articles' # Update this with the path to your articles folder
        articles = os.listdir(articles_dir)

        # Create a list of file names
        file_names = [f"Found file: {file_name}" for file_name in articles]

        # Return the list of file names
        return file_names
    else:
        # Return a default response if the command is not recognized
        return ['Sorry, I didn\'t understand that command.']
responses = [response for response in get_response('!post_feed')]
```

theDuke - March 5, 2023, 6:08 pm

So now it's posting all the articles at once? How many articles are you trying to display by the way? I could take a look at the code if you'd like and if you can show me an example of the output it's giving you on Discord I could see if I could help out in some way. I won't be on PC for a bit but I can still take a look without running it.

Mr. Robot - March 5, 2023, 6:09 pm

(Mar 5, 2023 07:08 PM)theDuke Wrote: [\[CITE\]](#) So now it's posting all the articles at once? How many articles are you trying to display by the way? I could take a look at the code if you'd like and if you can show me an example of the output it's giving you on Discord I could see if I could help out in some way. I won't be on PC for a bit but I can still take a look without running it.

I have updated the post to include everything needed to see how I am running it. Let me know if you can see an easy fix for it.

theDuke - March 5, 2023, 6:10 pm

(Mar 5, 2023 07:09 PM)Mr. Robot Wrote: [\[CITE\]](#) I have updated the post to include everything needed to see how I am running it. Let me know if you can see an easy fix for it.

Just saw! I'll take a look in a bit.

Mr. Robot - March 5, 2023, 7:04 pm

(Mar 5, 2023 07:10 PM)theDuke Wrote: [\[CITE\]](#) Just saw! I'll take a look in a bit.

Got it working with this:

bot.py

Code import discord

import responses

import asyncio

import os

import sys

import shutil

check if feed.txt file exists

if not os.path.exists('feed.txt'):

print('feed.txt file not found')

sys.exit()

```

# remove the articles folder if it already exists
if os.path.exists('articles'):
    shutil.rmtree('articles')

# create a folder to store the articles
os.makedirs('articles')

# read the feed.txt file
with open('feed.txt', 'r') as f:
    # split the file contents into articles
    articles = f.read().split('\n\n')

# iterate over each article and save it to a file
for i, article in enumerate(articles):
    # create a new file for the article
    filename = f'articles/{i+1}.txt'
    with open(filename, 'w') as article_file:
        # add ``` to the start of the file
        article_file.write('```\n')
        # write the article to the file
        article_file.write(article)
        # add ``` to the end of the file
        article_file.write('\n```')

async def send_message(message, user_message, is_private):
    try:
        response = responses.get_response(user_message)
        for article in response:
            await message.author.send(article) if is_private else await message.channel.send(article)
            await asyncio.sleep(2) # sleep for 2 seconds between sending each article

    except Exception as e:
        print(e)

def run_discord_bot():
    TOKEN = 'YOURTOKEN'
    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):
        if message.author == client.user:
            return

        username = str(message.author)
        user_message = str(message.content)

```

```

channel = str(message.channel)

print(f'{username} said: "{user_message}" ({channel})')

if user_message[0] == '?':
    user_message = user_message[1:]
    await send_message(message, user_message, is_private=True)
else:
    await send_message(message, user_message, is_private=False)

client.run(TOKEN)

if __name__ == '__main__':
    run_discord_bot()
main.py
Code import bot

if __name__ == '__main__':
    bot.run_discord_bot()
responses.py
Code import os
import random
from typing import List

def get_response(message: str) -> List[str]:
    p_message = message.lower()

    if p_message == '!post_feed':
        articles_dir = './articles' # Update this with the path to your articles folder
        articles = os.listdir(articles_dir)

        files = []
        # Create a list of file names
        # file_names = [f"Found file: {file_name}" for file_name in articles]

        for article in articles:
            with open(f'{articles_dir}/{article}') as file:
                if file.readable():
                    contents = file.read()
                    files.append(contents)
        # Return list of file contents
        return files
    else:
        # Return a default response if the command is not recognized
        return ['Sorry, I didn\'t understand that command.']

```