

Discord 通知 Bot

Discord 通知 Bot を作成する。通知 Bot は、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。

特徴

- 通知 Bot は、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。
- 通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。
- 通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。
- 通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。
- 通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。
- 通知 Bot は、Discord API を使用して、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。

インストール

依存関係

- `timer.py` - タイマー機能
- `ding.mp3` - 通知音
- `pomo.db` - データベース

インストール

```
pip install discord.py aiosqlite
```

FFmpeg のインストール

```
# Ubuntu/Debian
sudo apt install ffmpeg

# macOS
brew install ffmpeg

# Windows
# https://ffmpeg.org/download.html からダウンロード
```

使用方法

1. 設定

Discord 通知 Bot を起動する前に、Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。

ステップ 1: Discord サーバーに接続し、特定のチャンネルにメッセージを送信する。

```
export DISCORD_BOT_TOKEN='your_token_here'
```

ステップ2: 設定を永続化

```
# bash用
echo "export DISCORD_BOT_TOKEN='your_token_here'" >> ~/.bashrc
source ~/.bashrc

# zsh用
echo "export DISCORD_BOT_TOKEN='your_token_here'" >> ~/.zshrc
source ~/.zshrc
```

ステップ3: .envファイルの作成

```
# .env ファイルを作成
echo "DISCORD_BOT_TOKEN=your_token_here" > .env

# python-dotenvをインストール
pip install python-dotenv
```

2. 実行

```
python timer.py
```

実行結果:

```
実行結果
```

実行

実行結果

!pomo - 実行結果

実行:

```
!pomo [実行] [実行] [実行] [実行]
```

実行:

- 1. pomodoro- timer: 25
- 2. pomodoro- timer: 5
- 3. pomodoro- timer: 15
- 4. pomodoro-timer- timer: 4

例:

```
!pomodoro                    # 25分5分15分4分
!pomodoro 50                 # 50分
!pomodoro 50 10              # 50分10分
!pomodoro 50 10 20           # 50分10分20分
!pomodoro 50 10 20 3         # 50分10分20分3分
!pomodoro 1 1 1 2            # 1分2分
```

例:

- 1. pomodoro-timer
- 2. pomodoro-timer
- 3. pomodoro-timer
- 4. pomodoro-timer
- 5. pomodoro-timer
- 6. pomodoro-timer
- 7. pomodoro-timer
- 8. 4分pomodoro-timer
- 9. pomodoro-timer

!stats - 例

pomodoro-timer

```
!stats
```

例:

- pomodoro-timer
- pomodoro-timer

!test - 例

pomodoro-timer

```
!test
```

pomodoro

- 00000000 - 00000000000000000000000000000000
- ► 00 - 0000000000000000
- 000 - 00000000000000000000000000000000

1111

1. $\square \square \square \square \square \square \square \square - \square \square \square \square \square \square$
2. $\square \square \square \square \square \square \square \square \square \square - \square \square \square \square \square$

□ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □

□□□□□

1. pomodoro
- 2.
3. 1
4.
 -
 - ding.mp3
5.
 - 4 →
 - →
6.
 -
7. 2
- 8.

--	--	--	--	--	--	--	--

SQLiteデータベースpomo.dbデータベース

```
CREATE TABLE stats (  
  user_id INTEGER PRIMARY KEY,  
  total_minutes INTEGER DEFAULT 0,  
  sessions INTEGER DEFAULT 0  
)
```

- `user_id`: DiscordユーザーID
- `total_minutes`: 総プレイ時間
- `sessions`: セッション数

インストール

必要なパッケージ

1. FFmpeg

```
ffmpeg -version
```

2. ding.mp3

```
ls -l ding.mp3
```

3. テスト

```
!test
```

実行結果

4. データベース

- Discordサーバー
- データベース

5. タイマー timer.py

```
# timer.py 150秒間203回  
options='-filter:a "volume=2.0"' # 2.0 3.0 4.0
```

実行結果

1. 環境構築

```
echo $DISCORD_BOT_TOKEN
```

環境構築

2. Discord Developer Portal

3. 依存関係のインストール

```
pip list | grep discord  
pip list | grep aiosqlite
```

インストール

1. 依存関係のインストール

2. 依存関係のインストール

インストール

インストール

インストール

- インストール
- インストールFFmpeg
- インストールAPI
- インストール
- インストール

インストール

インストール

- インストール: インストール
- インストール: インストールGit
- .env: .env

```
# .gitignore  
echo ".env" >> .gitignore
```

インストール

1. Discord Developer Portal
2. インストール

3. 音量調整機能を実装する

音量調整機能を実装する

音量調整機能を実装する

timer.pyに以下のコードを追加する:

```
# 150Hzの音量調整機能を実装する
options='-filter:a "volume=2.0"'

# 203Hzの音量調整機能を実装する
options='-filter:a "volume=2.0"'
```

volume=2.0の値を変更する:

- 1.0 - 音量を100%にする
- 2.0 - 2倍
- 3.0 - 3倍
- 0.5 - 半減

音量調整機能を実装する

ding.mp3の音量調整機能を実装するtimer.pyにSOUND_FILEを追加する:

```
SOUND_FILE = "your_sound.mp3"
```

音量調整機能を実装する

音量調整機能を実装する:

```
async def pomo(ctx, work_minutes: int = 25, short_break: int = 5,
long_break: int = 15, long_break_interval: int = 4):
```

50分間作業する

```
async def pomo(ctx, work_minutes: int = 50, short_break: int = 10,
long_break: int = 20, long_break_interval: int = 4):
```

音量調整機能を実装する

音量調整機能を実装する

□□□: 2025□2□ □□□□□: 2.0