

# Créer un bot Discord pour nettoyer automatiquement les canaux d'un serveur.

Créer un bot Discord pour supprimer les messages de différents canaux est faisable avec Python et la bibliothèque `discord.py`. Voici un guide simple pour le mettre en place sous Debian.

---

## Étapes pour créer et exécuter le bot

### 1. Créer une application Discord

1. Rendez-vous sur le portail des développeurs Discord.
2. Cliquez sur **New Application**, donnez-lui un nom, et cliquez sur **Create**.
3. Allez dans l'onglet **Bot** et cliquez sur **Add Bot**. Confirmez.
4. Sous **Token**, cliquez sur **Copy** pour sauvegarder le token du bot (gardez-le privé).

### 2. Activer les intentions nécessaires dans le portail Discord :

1. Allez sur le portail des développeurs Discord.
2. Sélectionnez votre application (le bot).
3. Dans le menu de gauche, cliquez sur **Bot**.
4. Sous **Privileged Gateway Intents**, activez l'option **Message Content Intent**.
5. Cliquez sur **Save Changes**.

### 3. Inviter le bot sur votre serveur

1. Sous l'onglet **OAuth2 > URL Generator**, cochez :
  - `bot` dans **Scopes**.
  - Les permissions nécessaires dans **Bot Permissions** (par exemple, `Send Messages`).
2. Copiez l'URL générée et ouvrez-la dans votre navigateur.
3. Invitez le bot sur votre serveur Discord en suivant les instructions.

### 4. Obtenir les IDs des canaux

Activez le **Mode Développeur** dans Discord (Paramètres > Avancés > Mode Développeur). Faites un clic droit sur le canal concerné et sélectionnez **Copier l'identifiant**.

## 5. Installer les dépendances

Sur votre Debian, installez Python et les bibliothèques nécessaires :

```
sudo apt update
sudo apt install python3 python3-pip -y
pip3 install discord.py
```

## 6. Écrire le script Python

Créez un fichier `bot.py` avec le contenu suivant :

```
import discord
from discord.ext import commands
import asyncio

# Remplacez par votre propre token
TOKEN = "VOTRE_TOKEN_ICI"

# Remplacez par les IDs des canaux ou vous souhaitez effacer des messages
CHANNEL_IDS = [123456789012345678, 987654321098765432] # Exemple d'IDs de canaux

# Nombre maximum de messages a supprimer par execution
AMOUNT_TO_CLEAR = 100

# Initialisation du bot avec les intentions necessaires
intents = discord.Intents.default()
intents.messages = True # Permet au bot d'interagir avec les messages
intents.message_content = True # Necessary pour acceder au contenu des messages
bot = commands.Bot(command_prefix="!", intents=intents)

@bot.event
async def on_ready():
    print(f"Bot connected comme {bot.user}")

    for channel_id in CHANNEL_IDS:
        channel = bot.get_channel(channel_id)
        if channel:
            try:
```

```
# Purge des messages dans le canal
deleted = await channel.purge(limit=AMOUNT_TO_CLEAR)
print(f"{len(deleted)} messages supprimés dans le canal {channel_id}.")
except discord.errors.Forbidden:
    print(f"Accès refusé pour supprimer les messages dans le canal {channel_id}.")
except discord.errors.HTTPException as e:
    print(f"Erreur HTTP lors de la tentative de purge dans le canal {channel_id}: {e}")
else:
    print(f"Canal avec ID {channel_id} introuvable.")

# Arrêter le bot après l'exécution de la purge
await bot.close()

# Lancer le bot
bot.run(TOKEN)
```

## 7. Lancer le bot en arrière-plan avec `nohup`

Pour faire tourner le bot en arrière-plan, vous pouvez utiliser `nohup` pour qu'il continue à fonctionner même si vous fermez le terminal. Voici la commande à exécuter pour démarrer le bot et enregistrer les logs dans un fichier :

```
nohup python3 /path/to/bot.py > /path/to/bot_output.log 2>&1 &
```

Cette commande exécute le script en arrière-plan et redirige les sorties (logs et erreurs) vers le fichier `bot_output.log`.

## 8. Configurer une tâche cron pour redémarrer le bot chaque lundi à 5:00AM

Pour redémarrer automatiquement le bot chaque lundi à 5:00AM, vous devez configurer une tâche cron. Pour cela, éditez votre crontab :

```
crontab -e
```

Ajoutez la ligne suivante pour exécuter le bot chaque lundi à 5:00AM :

```
0 5 * * 1 nohup python3 /path/to/bot.py > path/to/bot_output.log 2>&1 &
```

`nohup` est optionnel puisqu'on n'exécute pas le script depuis le terminal.

Cela fera en sorte que votre bot soit exécuté automatiquement chaque lundi à 5:00AM.

# Conclusion

Avec ce guide, vous avez un bot Discord qui purge les messages tous les 7 jours dans les canaux de votre choix. Vous pouvez adapter le script et les paramètres selon vos besoins, et utiliser `cron` pour planifier l'exécution automatique du bot tous les lundis à 5:00AM.

## Points importants

- **Gardez votre token secret** : Ne partagez jamais votre token. Si vous pensez qu'il est compromis, régénérez-le depuis le portail Discord.
- **Permissions du bot** : Vérifiez que le bot a bien les permissions nécessaires sur chaque canal.
- **Testez localement** : Assurez-vous que le script fonctionne correctement avant de l'automatiser.

Revision #9

Created 24 November 2024 16:31:49 by Admin

Updated 24 November 2024 21:52:49 by Admin