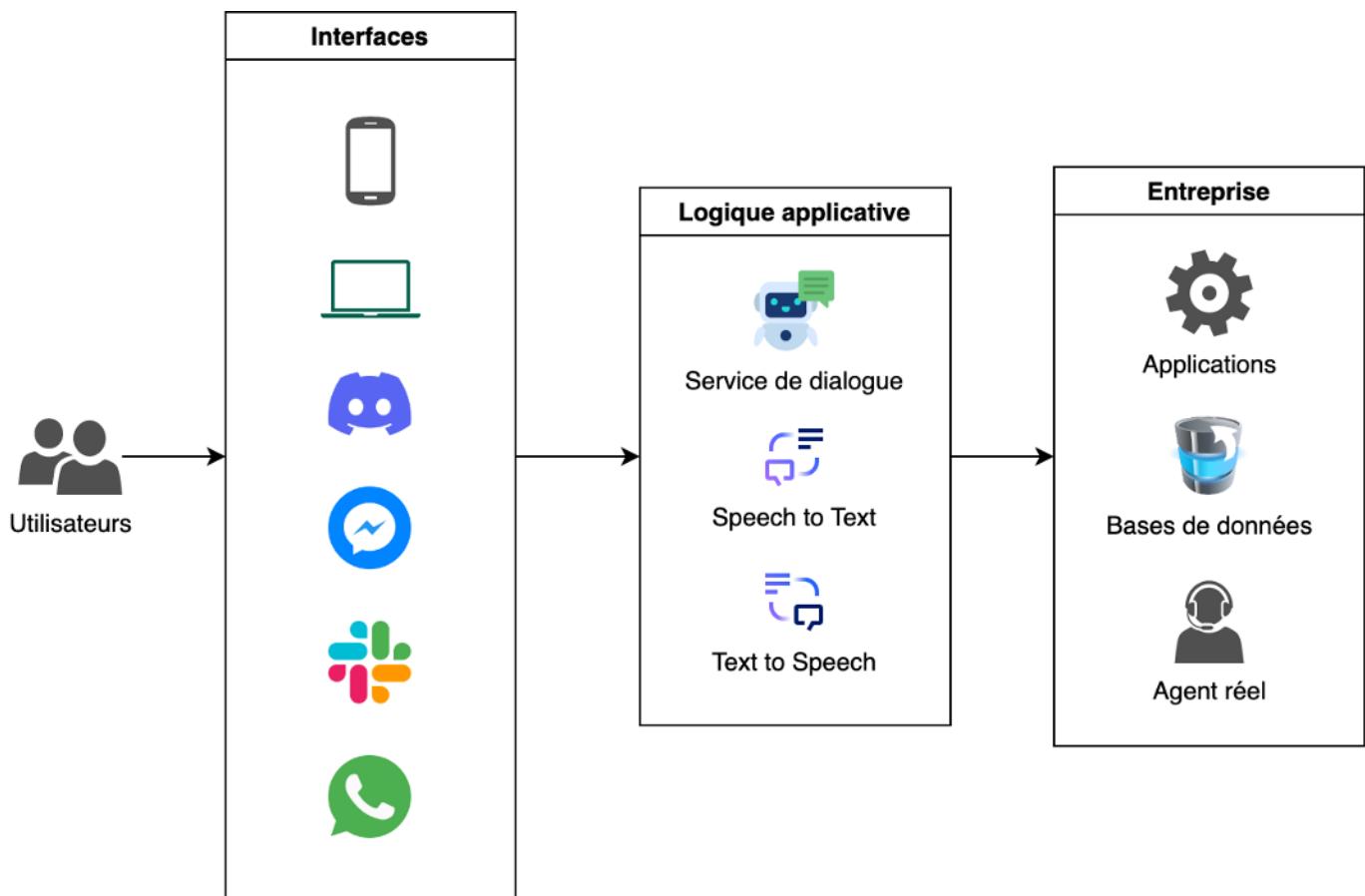


Intégration de votre Chatbot avec Discord

Discord est un logiciel propriétaire gratuit de VoIP et de messagerie instantanée. Il fonctionne sur les systèmes d'exploitation Windows, macOS, Linux, Android, iOS ainsi que sur les navigateurs web.

Conçu initialement pour les communautés de joueurs de jeux vidéo, son utilisation s'est diversifiée avec le temps et compte aujourd'hui plus de 300 millions d'utilisateurs.

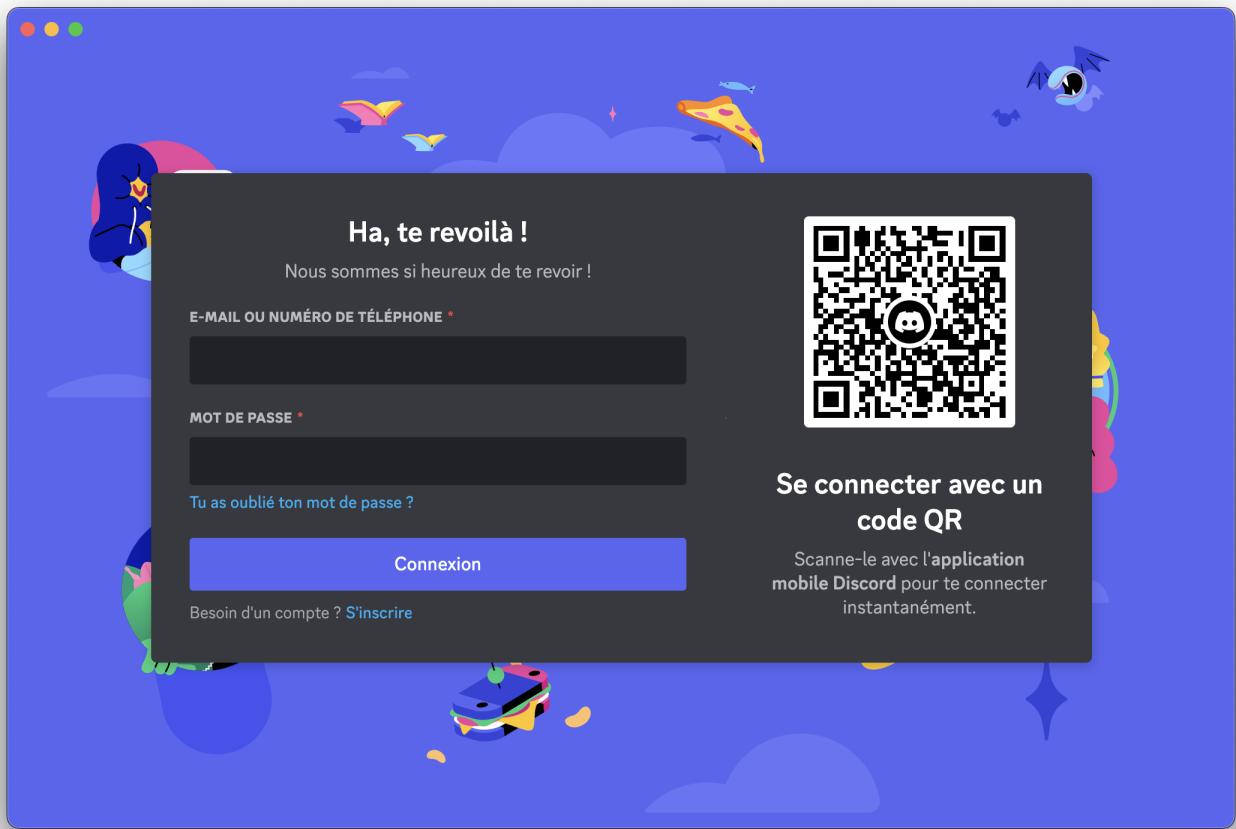
L'objectif de ce TP est de montrer un exemple d'intégration de Chatbot à une interface de communication, ici Discord. De nombreuses autres intégrations sont possibles, et fonctionnent de manière similaires pour tout logiciel de messagerie instantanée:



Accès Discord

Si vous n'avez pas Discord installé sur votre ordinateur, installez-le via ce lien: <https://discord.com/download>

Une fois Discord installé, connectez-vous à votre compte ou créez-en un nouveau:



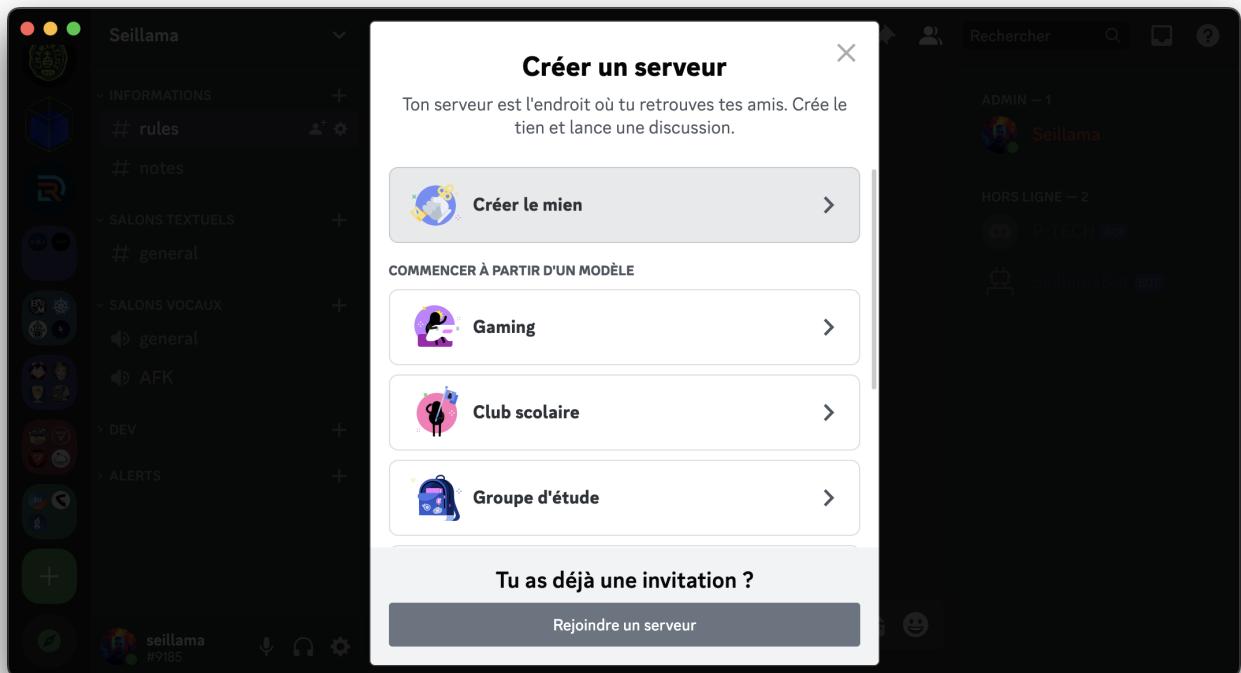
Serveur Discord

Pour permettre à votre chatbot the communiquer via un canal Discord vous aurez besoin d'un accès administrateur à un serveur Discord.

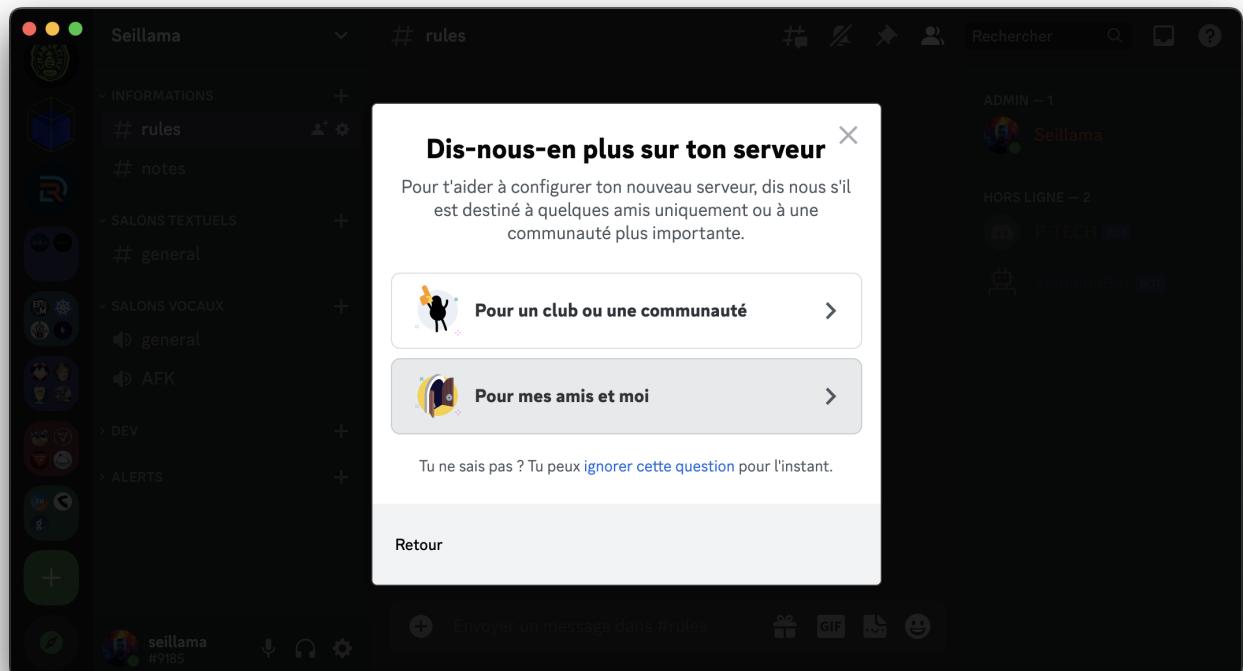
Créez un nouveau serveur Discord en cliquant sur **Ajouter un serveur** en bas à gauche de l'application:



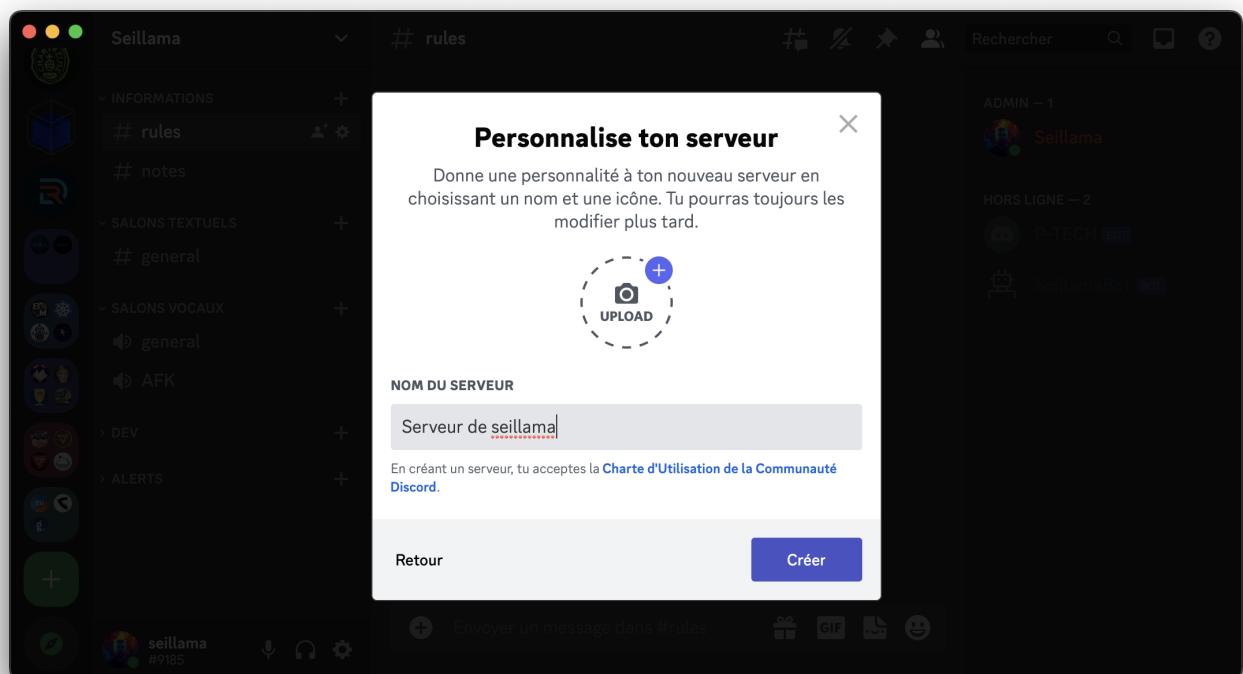
Selectionnerionner **Créer le mien**:



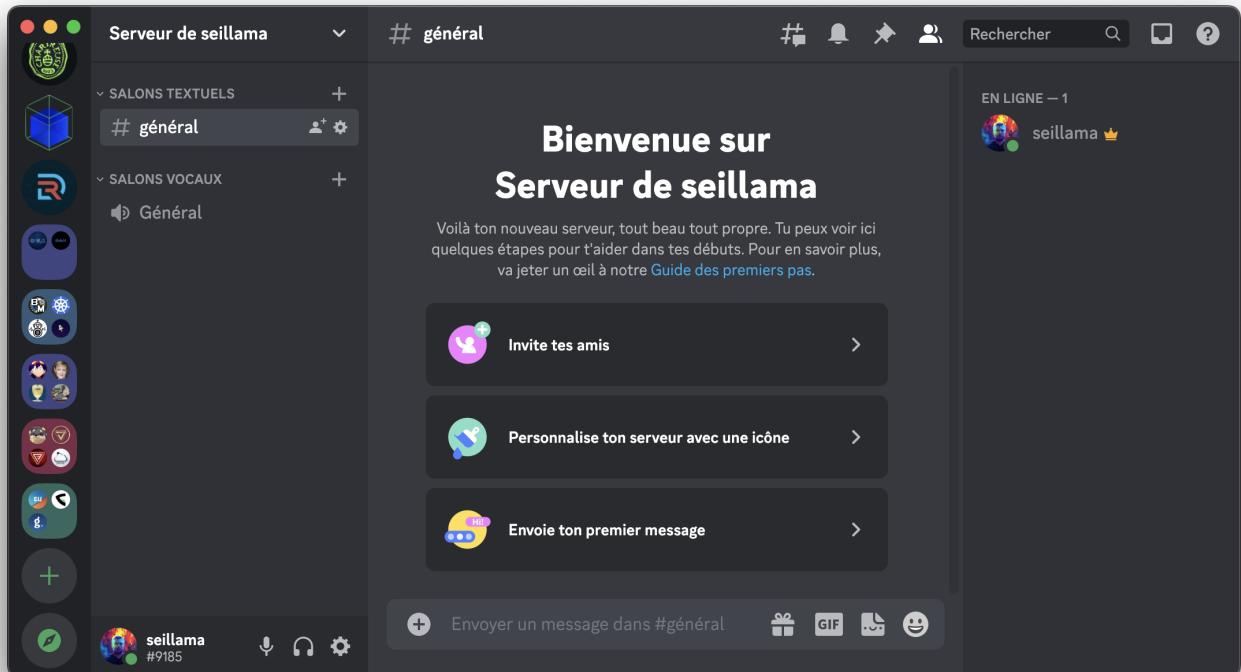
Selectionnerionner **Pour mes amis et moi**:



Saisir un nom de serveur:

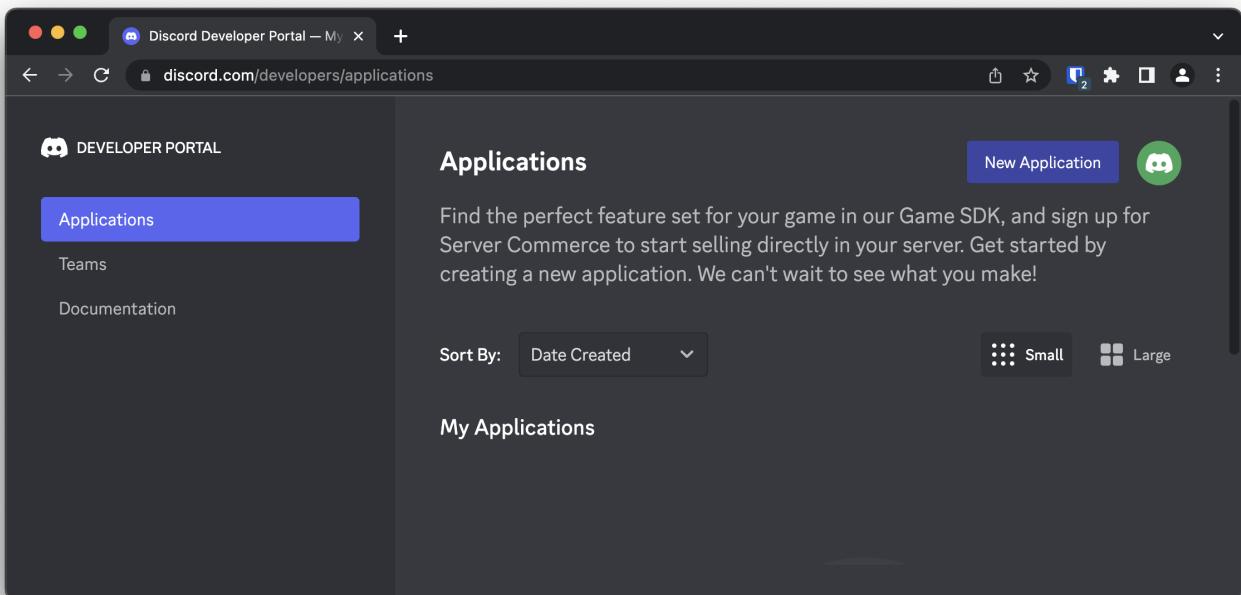


Et voilà! Votre serveur devrait être disponible!

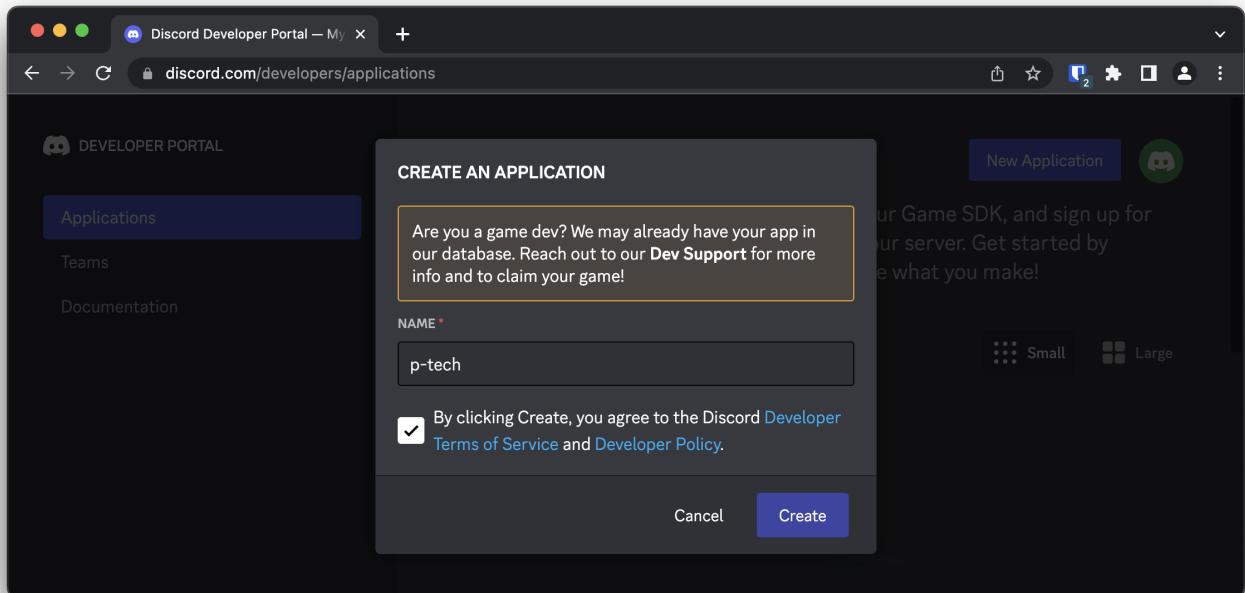


Application et Bot Discord

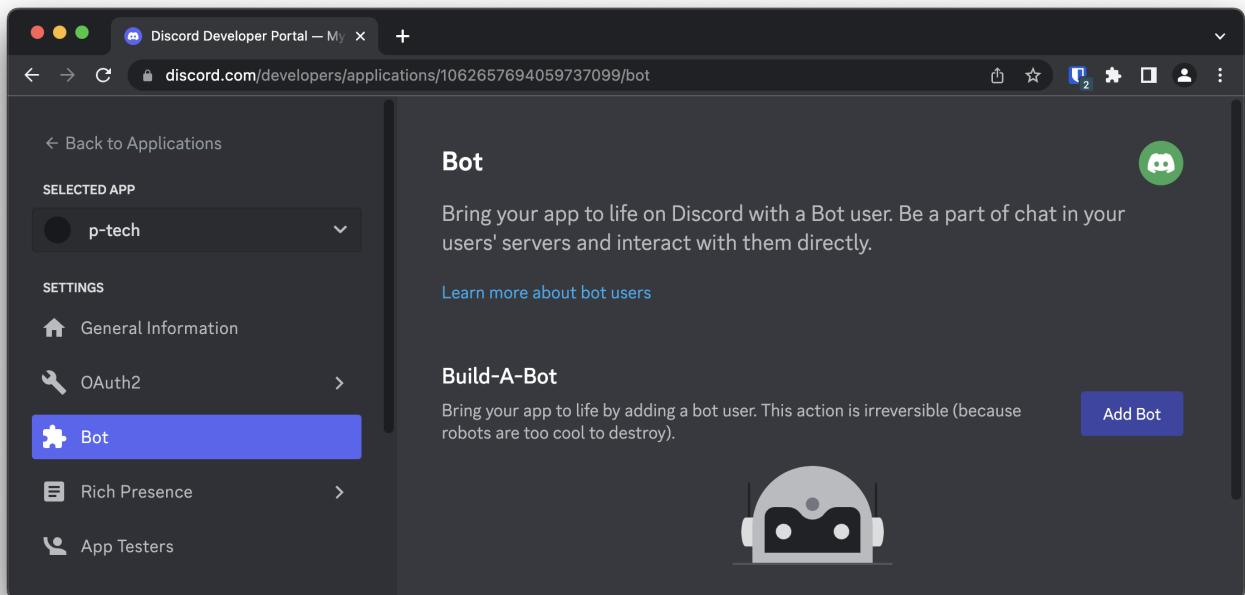
Une fois votre serveur créé et accessible, créez une nouvelle application discord en accédant à [Discord Developers](#):



Donner un nom à votre application et cliquez sur **Créer**:



Une fois votre application créée allez dans la section **Bot** et cliquer sur **Ajouter un bot**:



Une fois votre bot créé, activez **Message Content Intent** dans la partie **Privileged Gateway Intents** afin de permettre à votre bot de recevoir le contenu des messages discord:

The screenshot shows the Discord Developer Portal interface. On the left, there's a sidebar with a dropdown for 'SELECTED APP' set to 'p-tech'. Below it are sections for 'SETTINGS' (General Information, OAuth2, Bot, Rich Presence, App Testers), where 'Bot' is currently selected. The main content area is titled 'Privileged Gateway Intents'. It contains three sections: 'PRESENCE INTENT' (Required for your bot to receive Presence Update events, toggle switch off), 'SERVER MEMBERS INTENT' (Required for your bot to receive events listed under GUILD_MEMBERS, toggle switch off), and 'MESSAGE CONTENT INTENT' (Required for your bot to receive message content in most messages, toggle switch on). Each section includes a note about reaching 100 servers and links to 'Read more here'.

Toujours dans l'onglet Bot, cliquez sur **Réinitialiser le token**:

The screenshot shows the 'Build-A-Bot' section of the Discord Developer Portal. The sidebar is identical to the previous screenshot. The main content area is titled 'Build-A-Bot' with the sub-section 'TOKEN'. It features a large blue button labeled 'Reset Token'. Below the button, there's a note about tokens being irreversible for security reasons and a link to regenerate a new one. The 'Authorization Flow' section is also visible at the bottom.

Copiez le token généré et dans votre terminal Gitpod, et saisissez la commande suivante pour enregistrer le token dans une variable d'environnement **DISCORD_TOKEN** (celle-ci sera utilisée par la suite par votre script python pour connecter votre bot Discord):

```
> export DISCORD_TOKEN='MTA2Mj...2tEfQSw'
```

Inviter votre bot à votre serveur

De retour sur **Discord Developers**, allez à l'onglet **URL Generator** de la section **OAuth2**:

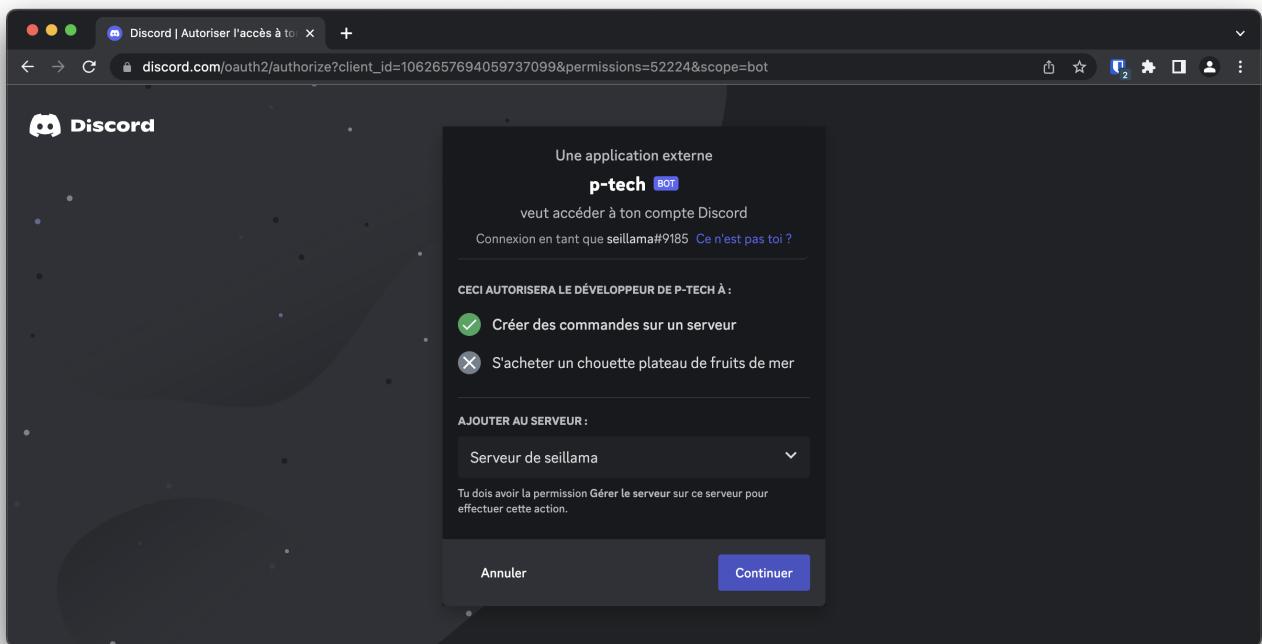
The screenshot shows the OAuth2 URL Generator page. On the left sidebar, under 'OAuth2', the 'bot' scope is selected. In the main area, the 'bot' scope is checked, while other scopes like 'identify', 'guilds', and 'applications.commands.permissions.update' are unchecked. A message at the top says: 'Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!'. There is also a green 'Copy' button.

Ajouter les permissions nécessaires à votre bot:

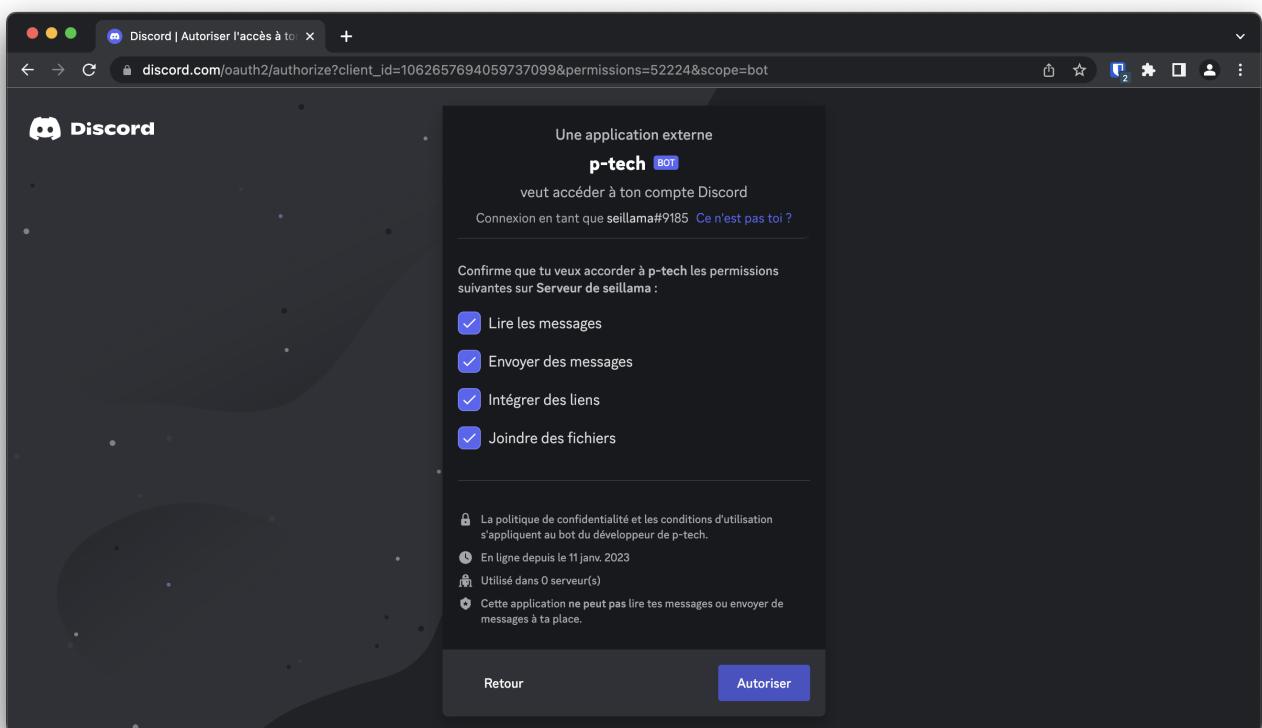
1. Selectionner le scope **bot**.
2. Selectionner **Read Messages/View Channels** dans **GENERAL PERMISSIONS**.
3. Selectionner **Send Messages, Embed Links, Attach Files** dans **TEXT PERMISSIONS**.
4. Cliquer sur **Copier** pour copier l'URL générée:

The screenshot shows the Bot Permissions page. Under 'BOT PERMISSIONS', the 'bot' permission is selected. In the 'TEXT PERMISSIONS' section, 'Send Messages', 'Embed Links', and 'Attach Files' are checked. In the 'VOICE PERMISSIONS' section, none are checked. At the bottom, the generated URL is https://discord.com/api/oauth2/authorize?client_id=1062657694059737099&permissions=52224&scope=bot, with a 'Copy' button next to it.

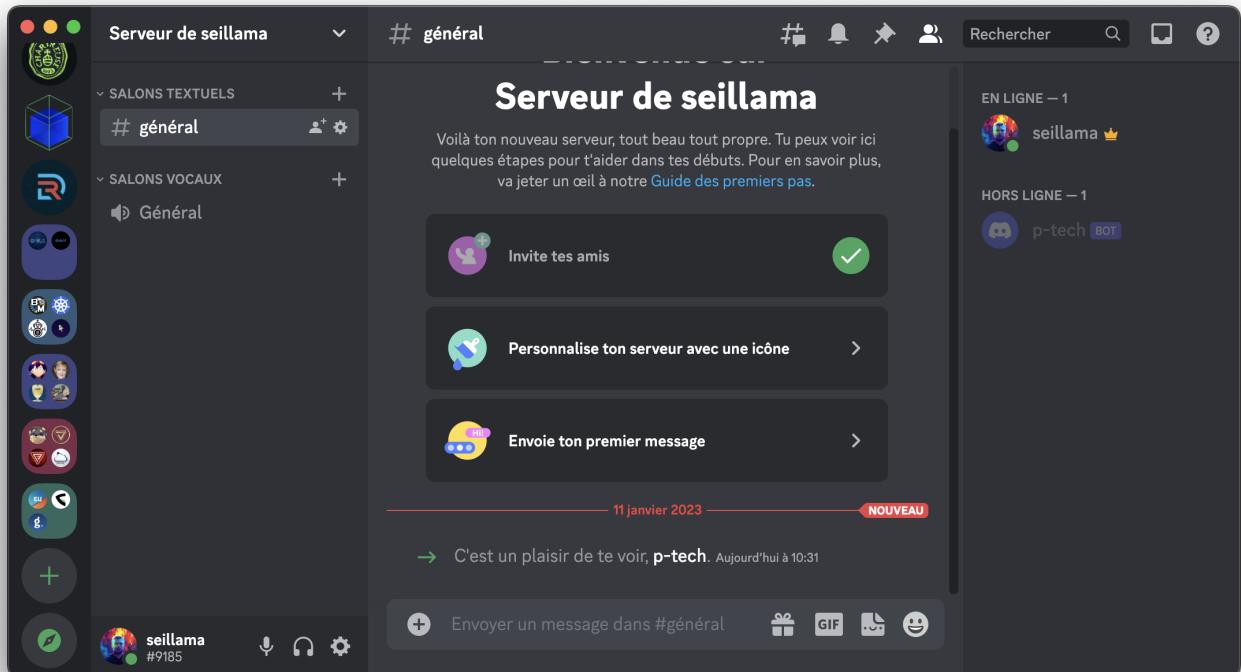
Ouvrez l'URL dans votre navigateur et inviter votre bot à votre nouveau serveur discord:



Authorizer les permissions demandées pour votre bot:



Votre bot devrait maintenant apparaître dans les utilisateurs de votre serveur!



Connecter votre bot

Après avoir ouvert votre code existant sur [Gitpod](#), saisir la commande suivante dans le terminal pour installer la librairie Python **discord**:

```
› pip install discord.py
```

Ensuite, créez un script **script-discord.py** pour ajouter l'intégration discord:

```
import discord
import os

# Init Discord client
intents = discord.Intents.default()
intents.message_content = True
client = discord.Client(intents=intents)

# Handles Discord login event
@client.event
async def on_ready():
    print(f'We have logged in as {client.user}')

# Run discord client using bot token
client.run(os.environ['DISCORD_TOKEN'])
```

Lancer le script python:

```
> script-discord.py
```

Vous deviez maintenant voir votre bot connecté à votre serveur:



Capturer les messages envoyés sur Discord

Modifiez votre script Python `script-discord.py` pour capturer les messages reçus sur le channel Discord, les afficher dans la console et les renvoyer dans le channel:

```
import discord
import os

# Init Discord client
intents = discord.Intents.default()
intents.message_content = True
client = discord.Client(intents=intents)

# Handles Discord login event
@client.event
async def on_ready():
    print(f'We have logged in as {client.user}')

# Handles Discord message event
@client.event
async def on_message(message):
    if message.author == client.user:
        return
    print(f'MESSAGE: {message.content}')
    await message.channel.send(f'REPONSE: {message.content}')

# Run discord client using bot token
client.run(os.environ['DISCORD_TOKEN'])
```

Lancer le script:

```
› script-discord.py
```

Essayez maintenant d'envoyer un message dans le channel **général** de votre serveur Discord, vous devriez voir ce message affiché dans la console de votre terminal ainsi que dans la réponse de votre bot:



Utilisez votre chatbot RiveScript

Modifiez votre script Python `script-discord.py` pour ajouter l'interpreteur RiveScript:

```
from rivescript import RiveScript
import discord
import os

# Init Rivescript bot
bot = RiveScript()
bot.load_directory("./eg/brain")
bot.sort_replies()

# Init Discord client
intents = discord.Intents.default()
intents.message_content = True
client = discord.Client(intents=intents)

# Handles Discord login event
@client.event
async def on_ready():
    print(f'We have logged in as {client.user}'')
```

```

# Handles Discord message event
@client.event
async def on_message(message):
    if message.author == client.user:
        return
    reply = bot.reply("localuser", message.content)
    await message.channel.send(reply)

# Run discord client using bot token
client.run(os.environ['DISCORD_TOKEN'])

```

Lancer le script:

```
› script-discord.py
```

Et voilà! Vous deviez maintenant pouvoir envoyer des messages à votre bot:

