

## 1.1°] Tutoriel : Installation de notre Bot

### Prérequis :

Vous devez posséder un compte Discord et un serveur.

### 1.1.1°] Installation du bot :

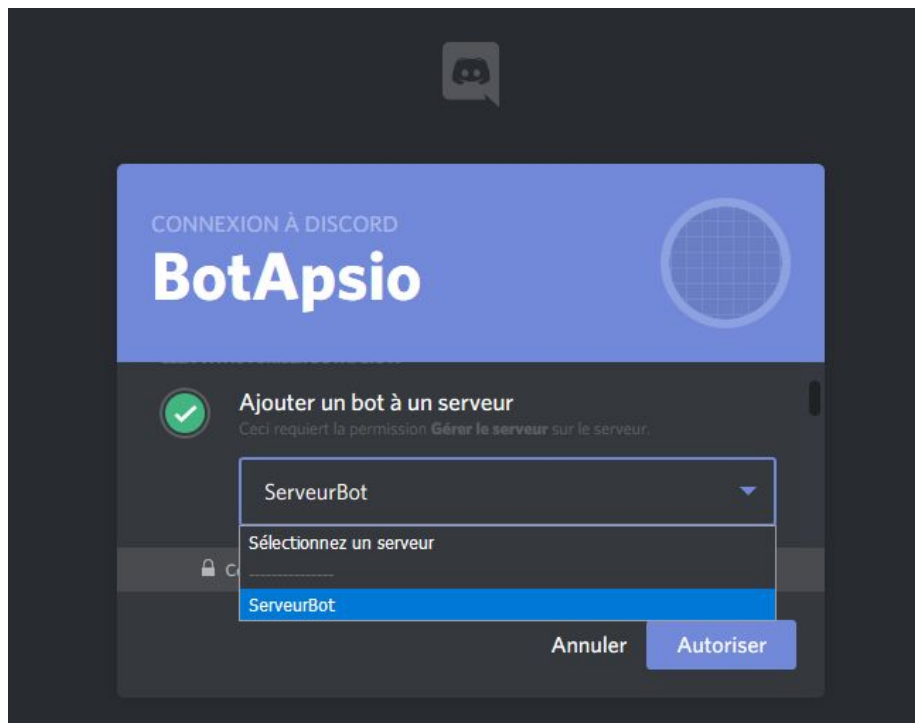
Cliquez sur ce lien ou copiez-le dans votre navigateur :

[https://discordapp.com/oauth2/authorize?client\\_id=552798530608300032&scope=bot&permissions=1141365953](https://discordapp.com/oauth2/authorize?client_id=552798530608300032&scope=bot&permissions=1141365953)

Vous arrivez sur cette page :

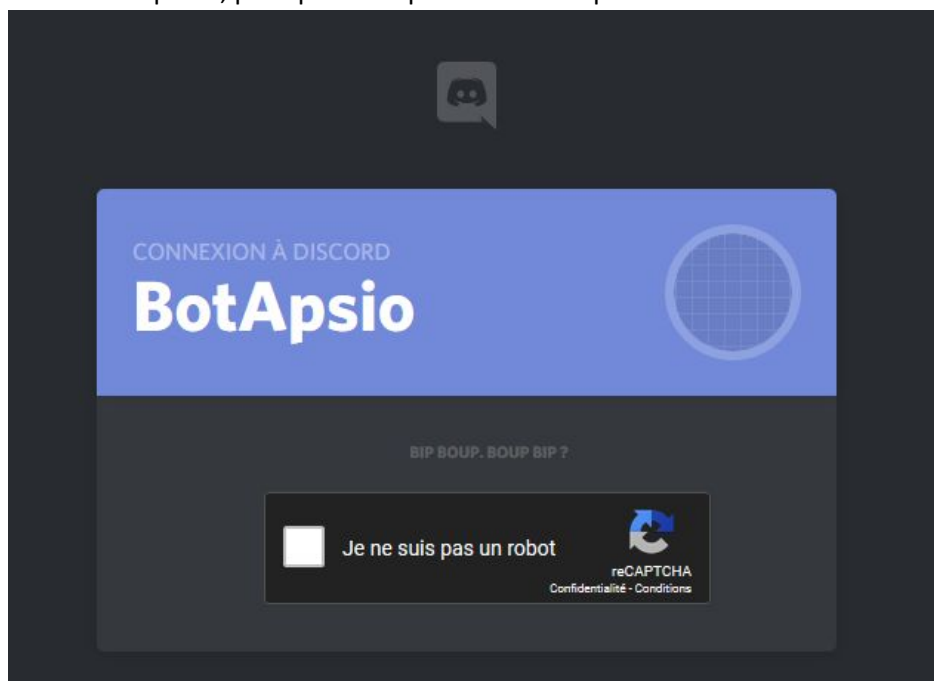


Choisissez sur quel serveur vous voulez installer le bot :

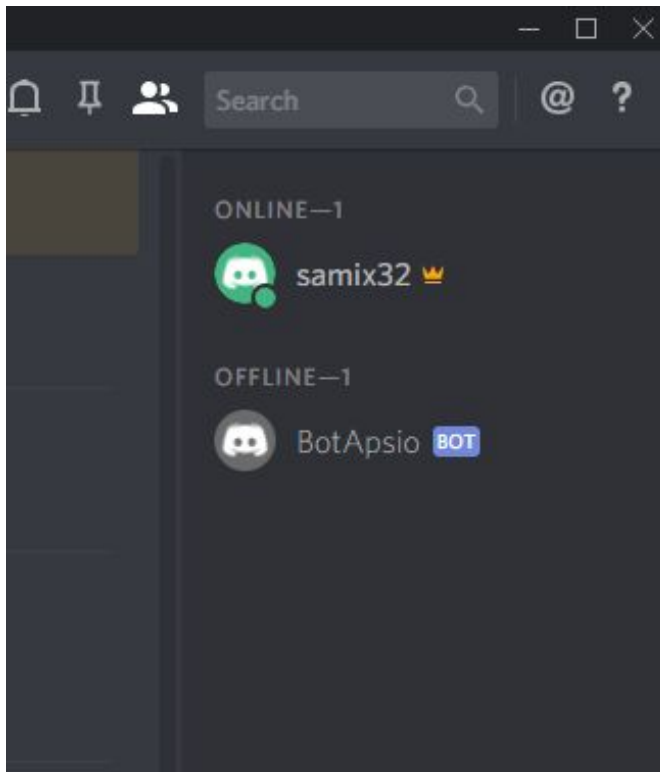


Puis cliquez sur Autoriser.

Validez-le captcha, pour prouver que vous n'êtes pas un robot



Si l'ajout du bot a bien été effectué, vous verrez le bot sur la liste des personnes qui sont sur votre serveur



### 1.1.2°] Installation du code

Le code du Bot, se trouve sur Framagit.

Faites un git clone de ce lien <https://framagit.org/jeremie.bourgeois/chatbot-discord.git>

## 1.2°] Utilisation du Bot

### 1.2.1°] Lancement du bot

Pour lancer le bot, ouvrez une invite de commande à partir du dossier où vous avez cloné le code du bot. Puis tapez la commande « `node index.js` ».

```
C:\Windows\System32\cmd.exe - node index.js
Microsoft Windows [version 10.0.17134.590]
(c) 2018 Microsoft Corporation. Tous droits réservés.

C:\Users\skyze\Documents\BOT Discord\bot>node index.js
```

Rendez-vous, ensuite sur Discord, où vous devrez apercevoir le bot connecté.



Vous pouvez maintenant commencer à l'utiliser

## 1.2.2°] Fonctionnalités

### 1.2.2.1°] Mode Apprentissage

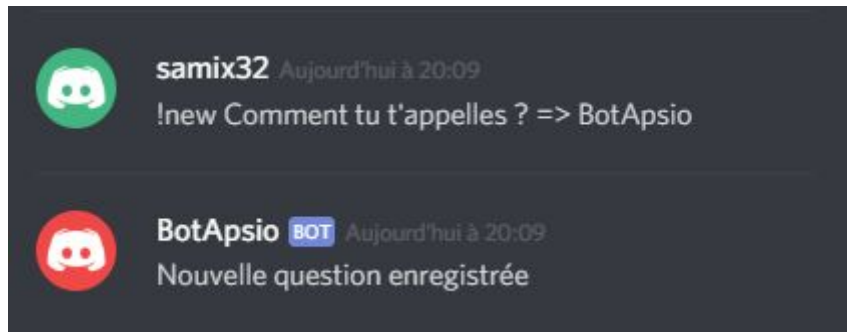
Le bot fonctionne avec une Base de données sous forme de fichier texte, qui contient les questions auxquelles le bot peut répondre.

Vous avez donc la possibilité de faire apprendre de nouvelles questions au bot. Pour cela, tapez '**!apprendre**' dans le channel sur Discord.



Comme vous pouvez le voir le bot informe l'utilisateur de la démarche à faire pour lui apprendre de nouvelles questions et réponses.

Exemple :



Comme on peut le voir, le Bot informe le bon enregistrement de la question.

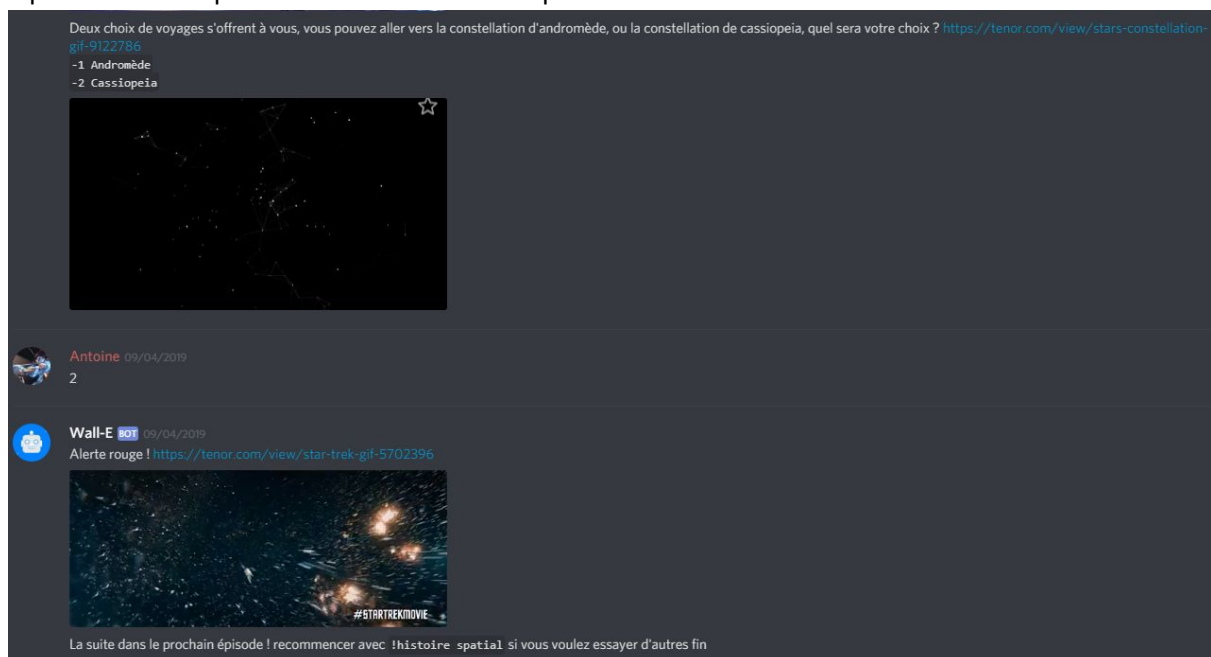
Teste de la nouvelle question :



### 1.2.2.2°] Mode Histoire

Pour les amateurs de séries Netflix, vous avez probablement déjà entendu parler des histoires où la fin dépend de nos choix, en effet cette fonctionnalité nous met en condition dans un univers où nos choix définissent la fin de cette histoire. Pour le moment, il existe une seule histoire et chaque choix est binaire, c'est à dire que soit on survit, soit c'est game over.

Le principe est simple, le bot pose une série de questions et il faut répondre avec 1 ou 2 (ou 3 et 4 s'il y avait plus de choix). Le bot attendra une réponse indéfiniment après avoir posé sa question mais au bout de quelques secondes il fera un rappel de comment répondre à sa question pour ne pas perdre l'utilisateur. De plus, il ne continuera son histoire uniquement si l'utilisateur répond avec 1 ou 2 car il filtre les messages reçus. Cela empêche d'avoir des réponses complètement incohérentes. Etant donné que le mode histoire peut être très répétitif car c'est un "die and retry" qui signifie que l'on doit recommencer de 0 plusieurs fois, on peut donner les réponses assez rapidement si on connaît les questions à l'avance.



Pour lancer le mode histoire il suffit d'entrer la commande **!histoire spatial**. Par la suite, on pourra rajouter d'autres histoire plus complexes les unes que les autres. On peut tâtonner le terrain en entrant une première commande **!** il va nous réorienter vers les différentes options (ici il indiquera juste que une commande **!histoire** existe mais si on fait simplement **!histoire** il va réorienter une nouvelle fois vers **!histoire spatial** car celle-là il la connaît)

## Explications :

```
//question 5
stop = false;
while (!stop) {
  try {
    let collected = await receivedMessage.channel.awaitMessages(reponse => reponse.content == "1" || reponse.content == "2", {
      max: 1,
      time: 20000,
      errors: ['time'],
    });
    stop = true;
    if (collected.first().content == '1') {
      receivedMessage.channel.send("Alerte noir ! https://tenor.com/view/uss-discovery-sporejump-spore-sporedrive-gif-10284882 ");
      receivedMessage.channel.send("La suite dans le prochain épisode ! recommencer avec `!histoire spatial` si vous voulez essayer d'autres");
    } else {
      receivedMessage.channel.send("Alerte rouge ! https://tenor.com/view/star-trek-gif-5702396");
      receivedMessage.channel.send("La suite dans le prochain épisode ! recommencer avec `!histoire spatial` si vous voulez essayer d'autres");
    }
  } catch(err) {
    receivedMessage.channel.send('Désolé `1` ou `2`');
  }
}
//fin itération 5
```

Typiquement, voilà à quoi ressemble une « question, réponse ».

1 - Dans un premier temps, on va créer une boucle infinie afin que l'utilisateur soit bloqué et qu'il ne doit donner que les possibilités que l'on veut, (par exemple « oui » ou « non »).

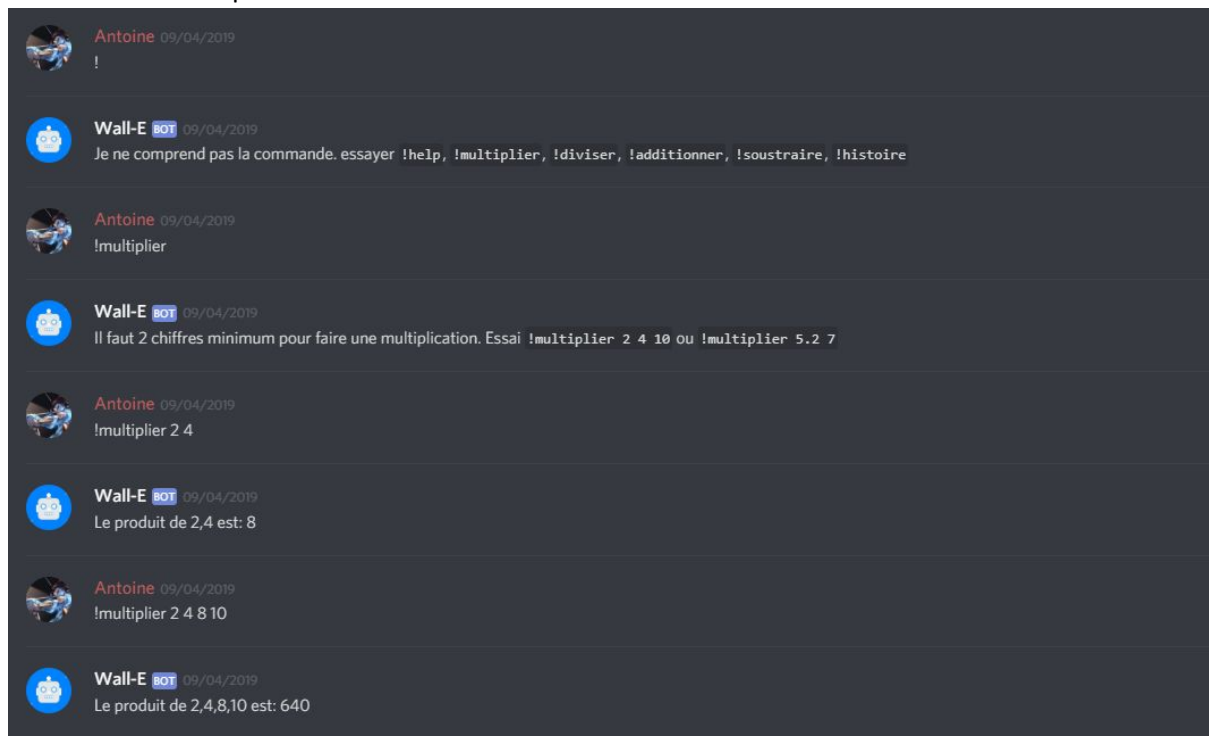
2 - Ensuite, on attend de la part de l'utilisateur une réponse, c'est ici que on va obliger le bot à poursuivre uniquement s'il reçoit « 1 » ou « 2 », on paramètre le temps d'attente du bot avant qu'il ne réaffiche un message, le nombre de message retenu si l'utilisateur envoie un string composé de 2 mots (il prendra que le premier) et comme c'est une « promesse » on prépare le catch de l'erreur (qui serait si l'utilisateur mettait 3)

3 - Enfin en fonction de ce que répondra l'utilisateur, on va afficher un message différent (qui pourrait être « vous avez gagné » ou « vous avez perdu »). On peut aussi complexifier l'histoire en imbriquant ce même bout de code dans le if de la réponse jugée bonne et en changeant les paramètres (ceux attendu par l'utilisateur, le temps avant que le bot réponde etc...)

### 1.2.2.3] Mode Calculatrice

Le mode calculatrice permet de faire des calculs simples comme une addition, une multiplication, une soustraction et une division. Pour ce faire il faut lancer la commande “!” qui va permettre d’orienter l’utilisateur. Ensuite plusieurs commandes seront affichées et notamment celles de calculs. Si l’utilisateur ne rentre pas les bons paramètres au moment de l’utilisation de la commande, un message est envoyé pour expliquer comment fonctionne ces commandes en particulier.

Etant une première ébauche de fonction, on ne peut pas encore coupler les opérations entre elles mais c’est tout à fait possible en enregistrant la variable résultat dans un fichier à l’image de la première fonctionnalité (celle de l’apprentissage) ainsi des opérations tel qu’une multiplication + une soustraction serait possible.



Même principe que précédemment, si on fait ! le bot nous indique que la commande !multiplier existe et si on fait !multiplier le bot nous indique que la commande !multiplier doit être précédée par des chiffres et montre un exemple de l’utilisation de la commande (exemple !multiplier 2 4 ).

### 1.2.2.4] Autres fonctionnalités

Le Bot possède d’autres petites fonctionnalités complémentaires. Il peut par exemple réaliser une recherche google à l’aide de la commande !google, vous donnez la météo à l’aide de la commande !meteo.