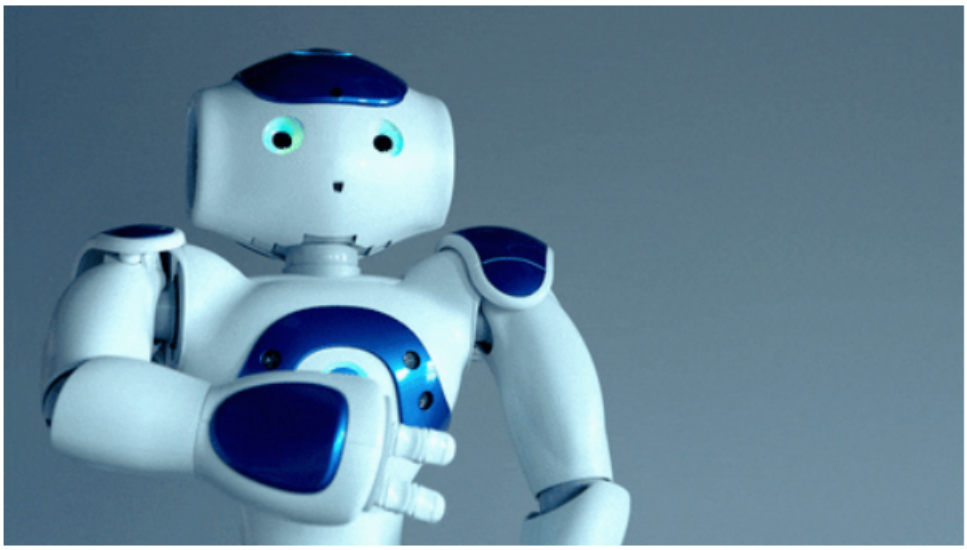


Compte rendu de projet Service Web
Chat Bot

7 juin 2022

TRAMONI Hadrien
CHIGNARD Julien

Table des matières

1 Installation et mise en œuvre

1.1 Téléchargement

Rendez-vous à l'adresse suivante : https://github.com/MrAntex/ChatBot et téléchargez l'archive en cliquant sur le bouton vert "Code" → *DownloadZip*.

1.2 Installation

Le projet est presque prêt, il manque les bibliothèques npm. Ouvrez un terminal dans le dossier ChatBot. Descendez dans le dossier Server et tapez : npm i. Revenez dans ChatBot et faites de même dans le dossier Bot. Remplacer le fichier dotenv vide par le dotenv contenant les différents tokens et autres données sensibles.

1.3 Lancement

Le projet est prêt. Positionnez-vous dans le dossier ChatBot et tapez : nodemon ./Server/app.js
Le terminal doit afficher : "Connected to db
Server listening to port 3000
Discord Bot is online ! "

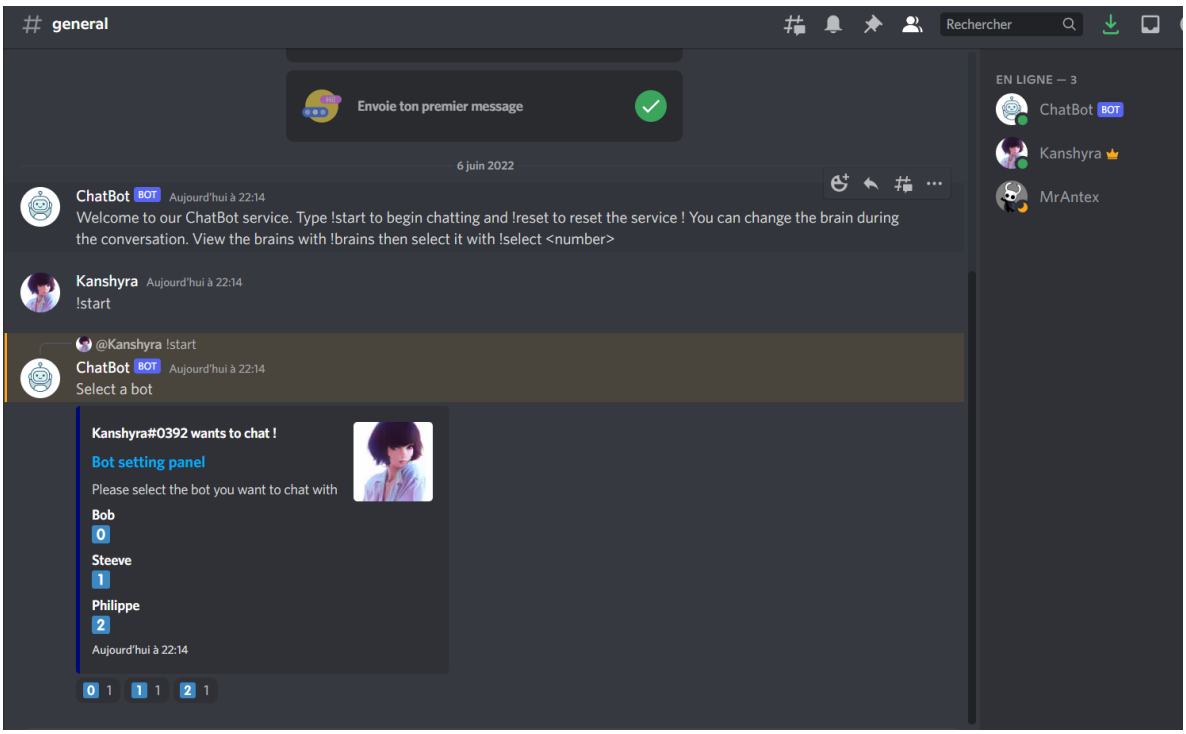
Vous pouvez discuter avec les bots de deux façons :

— par l'interface web à l'adresse http://localhost:port avec port une variable dans dotenv, fixée à 3000 par défaut.
— par l'interface Discord. Pour cela rejoindre le serveur : https://discord.gg/a2M5raVDkD

2 Guide d'utilisation

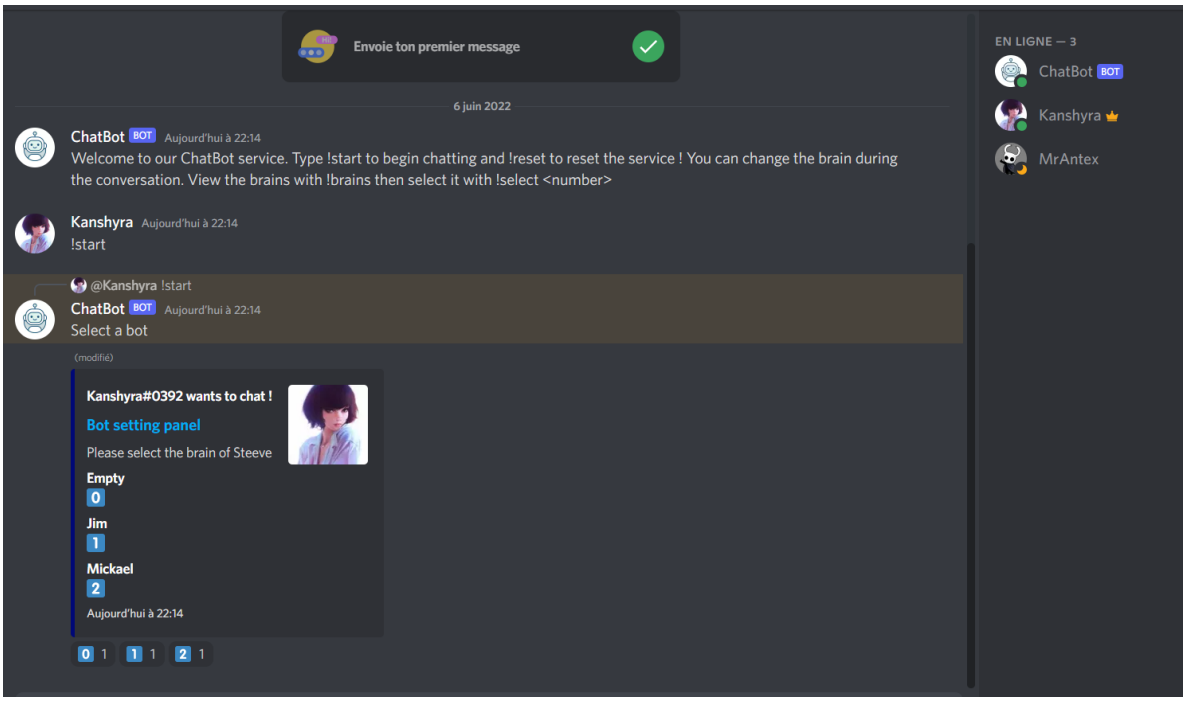
2.1 Interface Discord

Après avoir démarré le serveur principal, tapez dans le channel général : !start

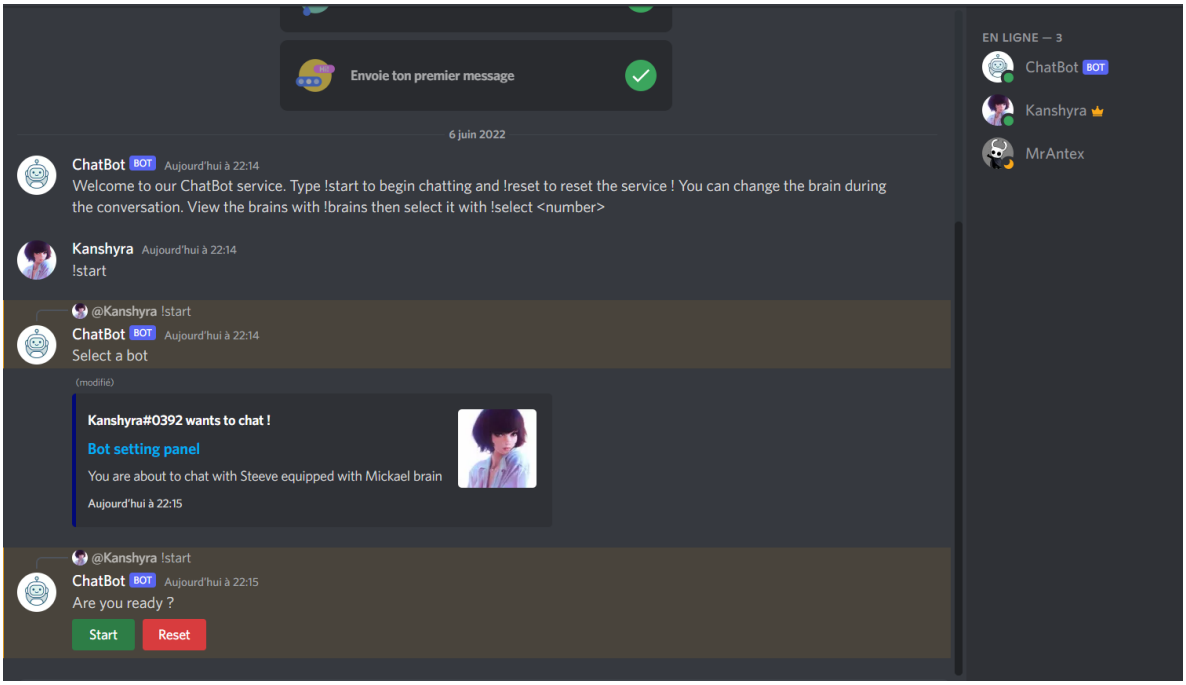


Il apparaît alors un embed. Dans ce dernier sont présents tous les bots ayant un accès à Discord. Pour en sélectionner un, il suffit de cliquer sur la réaction du numéro correspondant. ChatBot ne réagira qu'à la

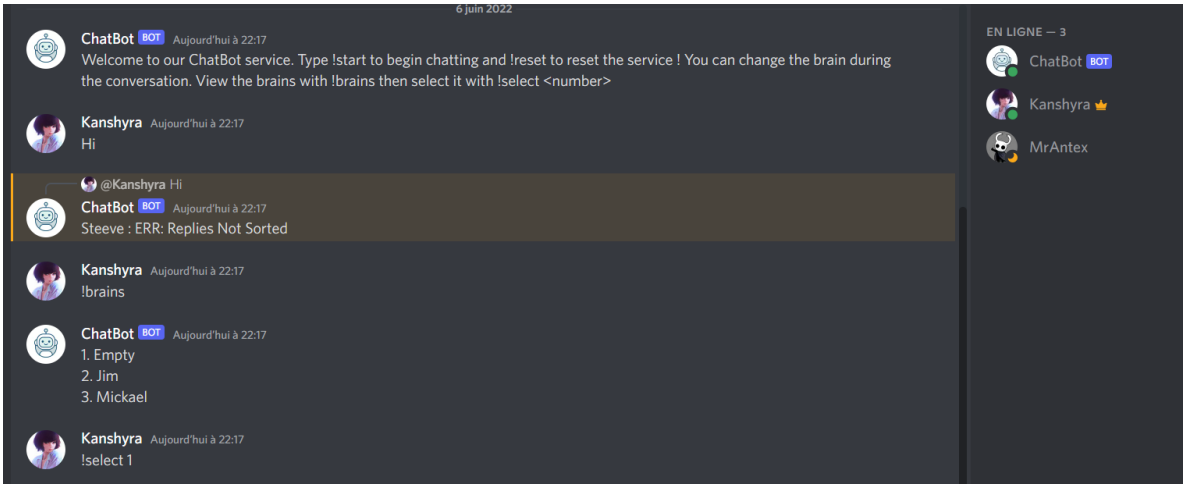
réaction de la personne ayant fait l'appel !start.



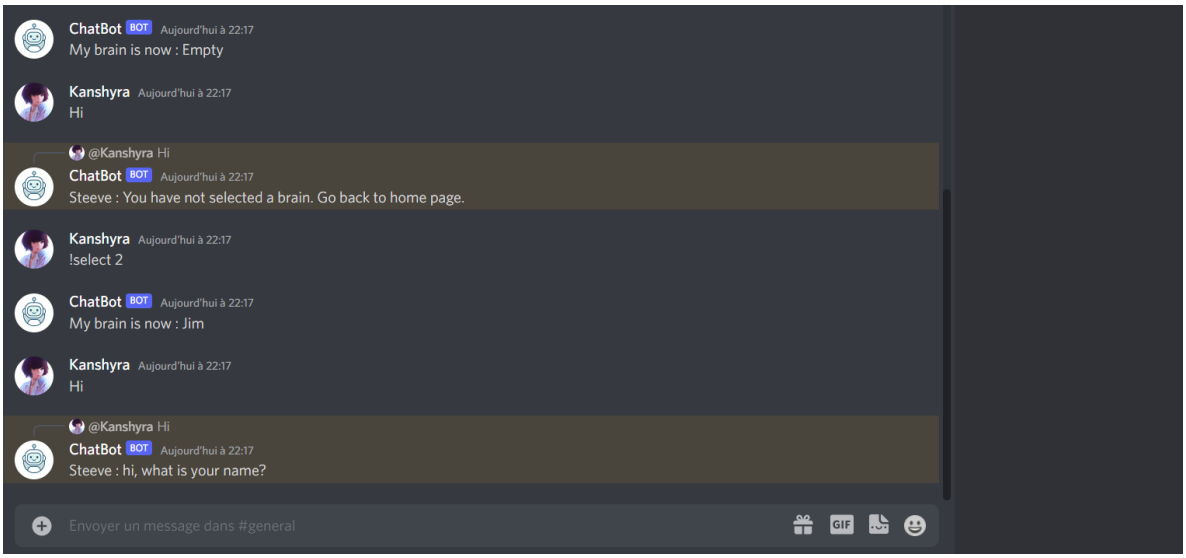
L'embed est actualisé. Il faut faire de même avec les cerveaux.
⚠La sélection du cerveau ne fonctionne pas malgré ce que le visuel laisse penser.



Appuyez sur Start pour démarrer le serveur du bot et discuter avec lui. Appuyer sur Reset pour reprendre à zéro ce processus.



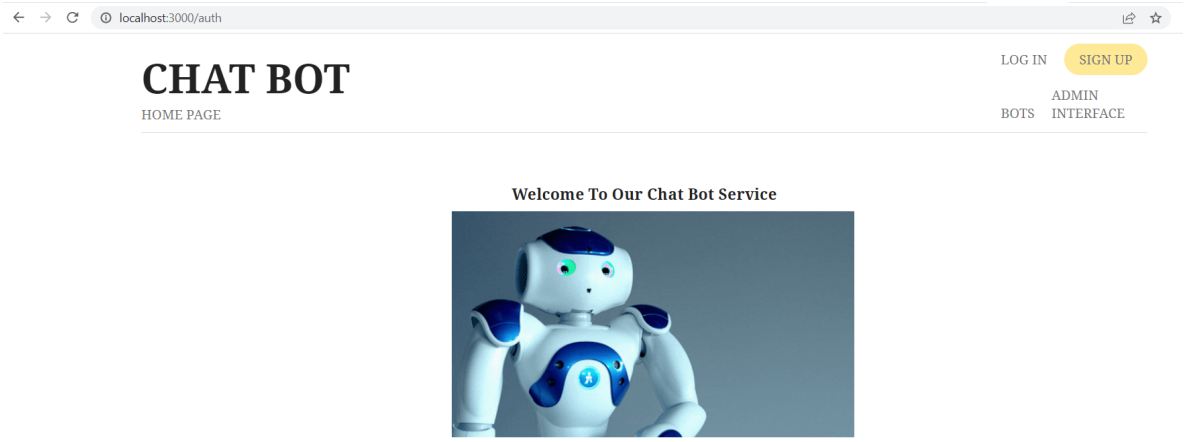
Comme on peut le voir ici, le bon bot a bien démarré, mais le cerveau n'a pas été chargé. En envoyant !brains vous pouvez voir la liste des cerveaux disponibles et en tapant !select <numéro>, le cerveau associé à ce numéro sera chargé. En utilisant cette même commande, il est possible de changer le cerveau durant la conversation.



2.2 Interface Web

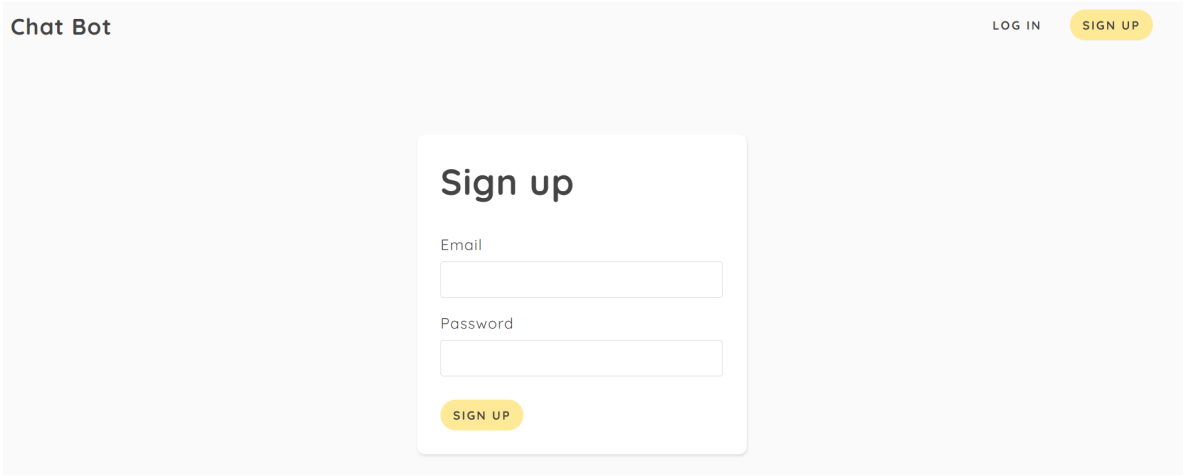
2.2.1 Côté Utilisateur

En accédant à l'URL http://localhost:port vous arrivez sur cette page d'accueil.



À partir d'ici, il vous est impossible d'aller plus loin sans vous authentifier. Essayer d'aller quelque part, vous serez redirigé sur la page de login.

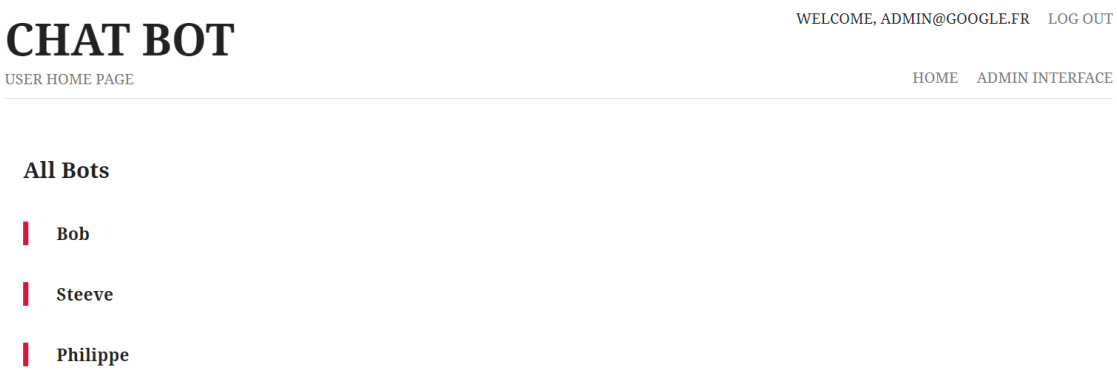
Supposons que vous êtes un utilisateur.



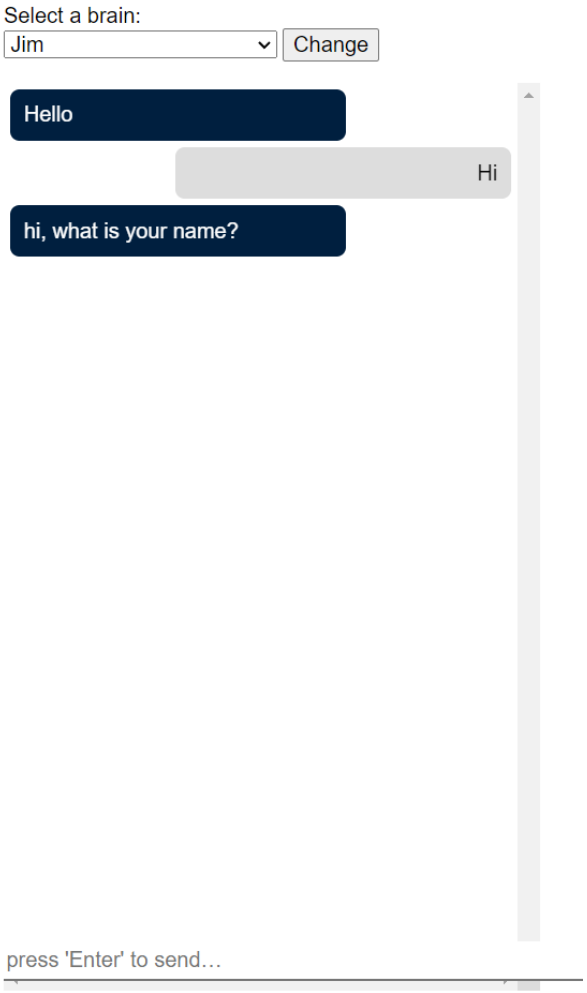
Entrez une adresse mail de format valide (l'existence de l'adresse n'est pas vérifiée, seule son format) ainsi qu'un mot de passe d'au moins 5 caractères. Cliquez sur SIGN UP et votre compte sera créé et ajouté à la base de données MongoDB. Vous serez automatiquement redirigé vers la page d'accueil et l'adresse mail de l'utilisateur connectée est rappelée en haut à droite.

Si vous ne désirez pas créer de compte, vous pouvez utiliser :
user : test@google.fr
password : azerty

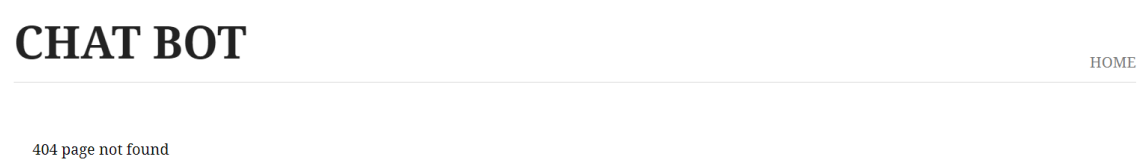
En cliquant sur Bots vous accédez à la page de sélection des bots.



Cliquez sur le bot avec lequel vous désirez converser et vous serez redirigés vers une page de dialogue.

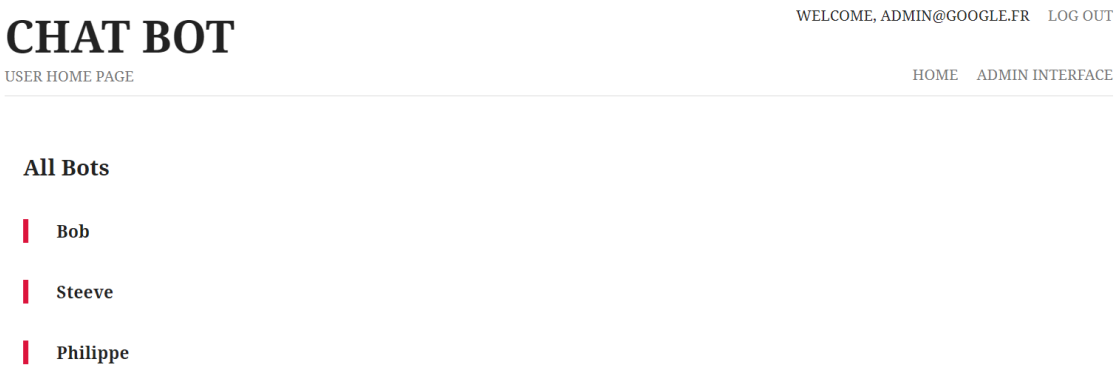


Utilisez le sélecteur pour choisir un cerveau et cliquez sur "Change" pour le charger. Cette fonctionnalité est capricieuse. Certaines fois elle permet de changer le cerveau pendant la conversation, d'autres fois elle ne fonctionne pas malgré que la console affiche avoir rechargé le cerveau.

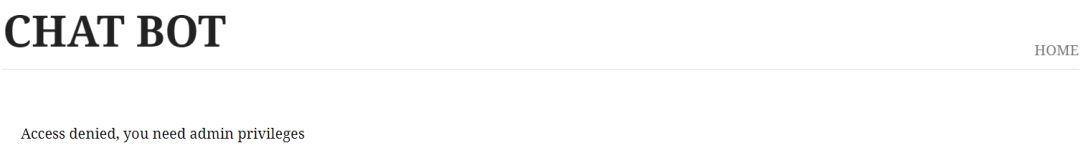


Si un utilisateur tente d'accéder à une URL non définie, il est redirigé vers une page 404.

2.2.2 Côté Administrateur

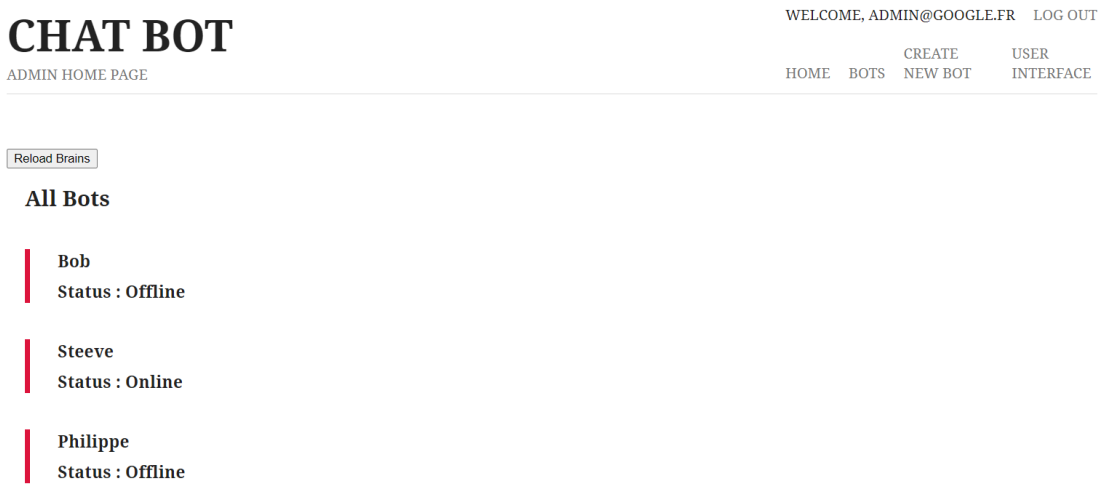


Depuis la page d'accueil ou de sélection des bots, il est possible de cliquer sur un onglet "Admin interface". Dans le cas où l'utilisateur connecté n'aurait pas les privilèges administrateurs, il est redirigé vers cette page.

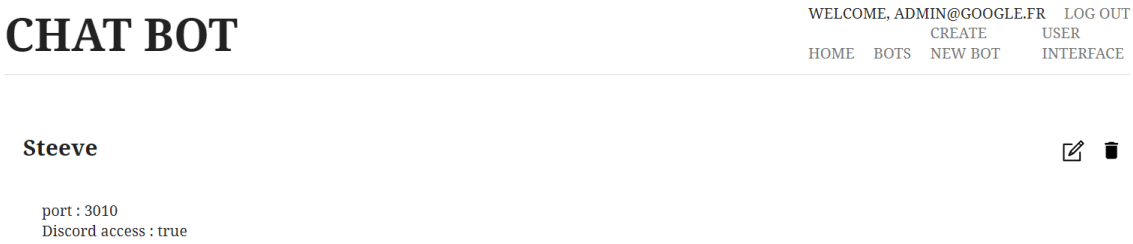


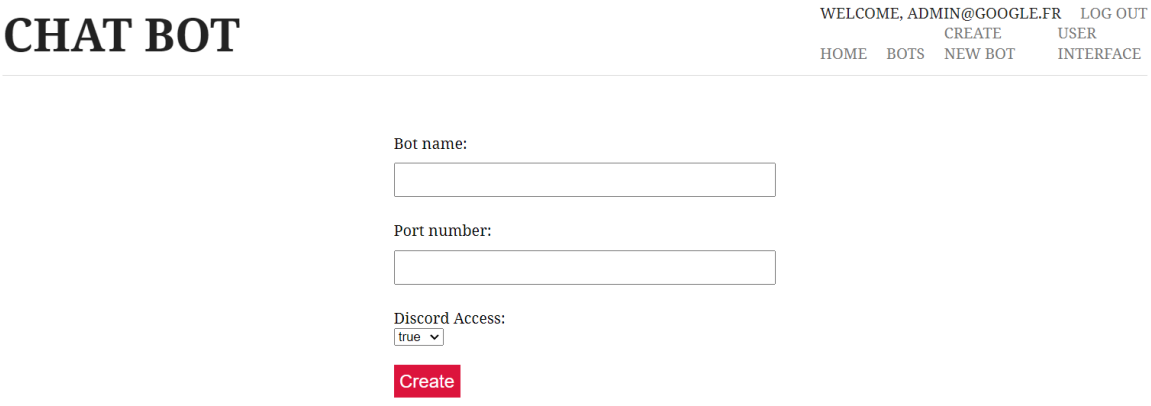
Il n'y a actuellement qu'un seul compte administrateur :
user : admin@google.fr
password : admin

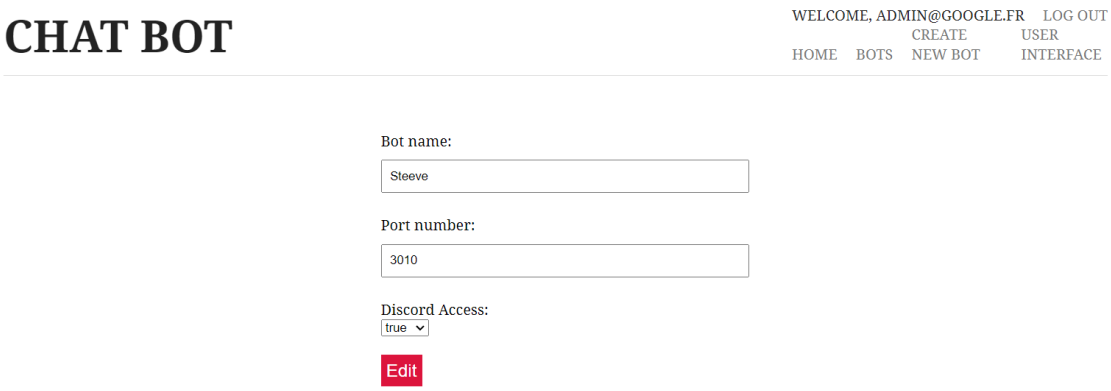
Pour donner l'accès administrateur, il n'y a qu'un seul moyen : modifier dans MongoDB le booléen "privilege" d'un compte.



Depuis l'interface admin, nous pouvons voir quels bots sont actuellement en ligne. En cliquant sur un bot, il est possible de voir ses propriétés, les éditer ou supprimer le bot. Le bouton "Reload Brains" vide la base de données des cerveaux et la remplit de nouveau avec le contenu du dossier ChatBot/Server/brains.







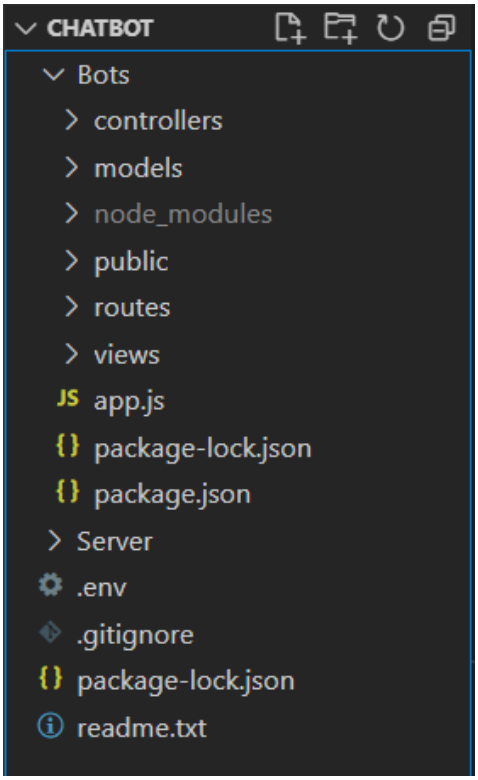
La création et l'édition d'un bot sont assez similaires.

3 Points techniques

3.1 Authentification

Lors de la création d'un compte, le mot de passe est hashé. Lors de la connexion, un token JWT est stocké dans les cookies. Si le token est valide, le cookie permet de garder connecté l'utilisateur entre les pages.

3.2 Architecture du projet



Nous avons structuré notre projet en deux parties. La partie Bot qui contient tout le fonctionnement d'un seul bot avec son propre processus et la partie Serveur principal qui héberge toute la partie interface administrative. Ces deux parties sont construites selon une architecture MVC. Les Modèles sont les fichiers de structurations des documents MongoDB, les Views sont tous les visuels affichés. La partie contrôleur a

été scindée en une partie "route" qui affecte à un type de requête sur une certaine route la fonction à utiliser et une partie contrôleur qui contient toutes ces fonctions.

4 Points à améliorer

4.1 REST API

Nous avons rencontré un problème lors de l'implémentation de l'interface Discord. Nous avons configuré les routes de sorte que l'authentification est vérifiée avant d'autoriser chaque requête. Ainsi, l'interface web est REST et sécurisé. Le problème est le suivant : pour rester REST, il faut que l'interface Discord soit connectable à notre projet simplement avec les requêtes HTTP, sans qu'il y ait de modifications à faire. Or il n'y a pas d'authentification côté Discord donc les requêtes sont toutes filtrées. Pour remédier à cela, nous avons fait une copie des fonctions déjà codée dans un contrôleur dédié à Discord qui lui n'a pas besoin d'authentification. Pour rester sur une architecture REST, il aurait fallu créer un utilisateur fictif qui endosserait toutes les requêtes de Discord.

4.2 Base de données

Nous n'avons pas réussi à ce que l'interface Discord récupère les données depuis la base de données. C'est à nouveau un problème pour une architecture REST. Nous réussissons bien à récupérer les bots par la BDD mais les cerveaux sont récupérés d'une façon détournée alors qu'ils devraient être facilement récupérable de la même façon que les bots.

4.3 Indépendance des processus

Lorsque l'on veut dialoguer avec un bot, le serveur principal lance avec la commande spawn un nouveau processus qui fait vivre le bot. Ce nouveau processus est censé être indépendant grâce à l'argument detached :true mais il semblerait que la fermeture du processus du Serveur principal entraîne l'arrêt des processus bots.

4.4 Arrêt d'un bot

Une fois lancé, un bot ne s'arrête plus. Il aurait fallu voir en temps réel s'il y avait encore une personne présente sur la page. Dans le cas contraire, kill le processus du bot par PID.

4.5 Utilisation simultanée Web/Discord

Puisque ces deux interfaces communiquent avec les mêmes bots, le changement d'un cerveau par une interface se répercute sur l'autre et cela peut causer des problèmes.

5 Conclusion

Voici le récapitulatif des Use Cases qui ont été traité.

Use cases	Complétion
Créer bot avec une interface	
Communiquer avec une autre interface grâce à une adresse et un port	
S'authentifier	
Sélectionner un cerveau et l'ajouter au bot	
BONUS : changer de cerveau pendant une conversation	
Le bot garde en mémoire des informations	
BONUS : Stockage par MongoDB	
Sélectionner un fichier de paramètres de connexion à Discord	
BONUS : connexion à Slack	
Supprimer l'accès à Discord d'un bot	
A travers une interface, voir l'état des différents bots	
BONUS : voir les conversations dans une interface	
BONUS : recouper des informations entre les bots	