

*documentation  
du projet*



Ce projet consiste à créer un bot Discord capable de traquer les messages envoyés dans un serveur Discord lorsqu'il est connecté, de les stocker dans une base de données, et de les rediffuser via des commandes spécifiques.

## ***Créer une Application Discord et Obtenir un Token de Bot***

Pour commencer, j'ai téléchargé l'application Discord et créé un compte. Ensuite, j'ai accédé au Discord Developer Portal pour créer une application et un bot Discord, afin de copier le token nécessaire pour authentifier le bot.

## **Configuration de l'Environnement de Développement**

Pour configurer votre environnement de développement, j'ai commencé par télécharger et installer Node.js, ce qui installe également npm. Ensuite, j'ai ouvert mon terminal et exécuté les commandes suivantes pour créer un nouveau dossier et initialiser un projet Node.js :

```
//Cette commande crée un nouveau dossier nommé "discord-bot"
mkdir discord-bot
cd discord-bot
//Cette commande initialise un nouveau projet Node.js dans le
dossier actuel ("discord-bot")
npm init -y
```

J'ai installer le package discord.js pour interfacier avec Discord

```
npm install discord.js
```

# Configurer le Bot

## 1. Créer un Fichier de Configuration :

J'ai crée un fichier nommé config.js dans mon projet :

```
module.exports = {
  token:
    "MTIzNzM2NTgxNTgyMTU5ODc1MA.GAmmd-.oBi9_6s5BtzKiGXl
    MEB6cHayeHGCS0auX2ox5U",
  prefix:"/"
}
```

## 2. Créer le Fichier Principal main.js :

J'ai crée un fichier nommé main.js et ajoute le code suivant pour configurer le bot et le connecter à Discord et les commandes :

```
const discord = require('discord.js');
const { Client, GatewayIntentBits } = require('discord.js');
const channel = client.channels.cache.get('894700086242734103');

const bot = new Client({ intents: [GatewayIntentBits.Guilds,
GatewayIntentBits.GuildMessages] });
const config = require("./config")

const { formatMessage, formatChannelMessages } =
require('./formatMessage');
const { getLastNMessages, getLastNMessagesChannel,
getMostActiveUser } = require("./db");

bot.commands = new discord.Collection();

bot.login(config.token);

channel.send(messageContent)
  .then(() => console.log('Message envoyé avec succès'))
  .catch(error => console.error('Erreur lors de l\'envoi du message :',
error));

bot.on("ready", async () => {
  console.log(`${bot.user.tag} est bien en ligne`);
});
```

```
bot.on("messageCreate", async (message) => {
  console.log("in message")
  if (message.author.bot) return;
  if (!message.content.startsWith(config.prefix)) return;

  const args = message.content.slice(config.prefix.length).trim().split(/
+/);
  console.log("args",args)
  const commandName = args.shift().toLowerCase();

  const command = bot.commands.get(commandName);

  if (!command) return; // Ignore les commandes non reconnues

  try {
    await command.execute(message, args);
  } catch (error) {
    console.error(error);
    message.reply("Une erreur s'est produite lors de l'exécution de la
commande.");
  });
});

bot.commands.set("/hey", {
  execute: async (message) => {
    await message.channel.send("Hey❤️");
  });
});

bot.commands.set("/who-am-i", {
  execute: async (message) => {
    const userInfo = `You are: ${message.author.username}\nYour id is:
${message.author.id}\nYour avatar is: ${message.author.avatarURL()}\nThis
server id is: ${message.guild.id}`;
    await message.channel.send(userInfo);
  });
});

bot.commands.set("/show-message", {
  execute: async (message, args) => {
    const n = parseInt(args[0]);
    const username = args.slice(1).join(" ");
  });
});

bot.commands.set("show-message-channel", {
  execute: async (message, args) => {
    const n = parseInt(args[0]);
    const channelId = args[1];
  });
});
```

```
bot.commands.set("talk-too-much", {
  execute: async (message, args) => {
    const channelId = args[0];

    // Vous devez obtenir vos messages ici et les stocker dans une
    // variable messages
    const messages = []; // Remplacez cela par votre méthode pour
    // obtenir les messages

    const formattedMessages = messages.map(formatMessage).join(' ');
    message.channel.send(formattedMessages);
  }

  if (message.content === '!format-channel-messages') {
    // Vous devez obtenir vos messages de canal ici et les stocker
    // dans une variable channelMessages
    const channelMessages = []; // Remplacez cela par votre méthode
    // pour obtenir les messages de canal

    const formattedChannelMessages = channelMessages.map(
      formatChannelMessages(channelMessages);
    message.channel.send(formattedChannelMessages);
  }
});
```

### 3. Créer un fichier Bases données db.js

Créez un fichier db.js pour gérer la connexion à la base de données:

```
const mysql = require('mysql');

const dbConfig = {
  host: 'localhost',
  user: 'ouss',
  password: 'ouss',
  database: 'discord'
};

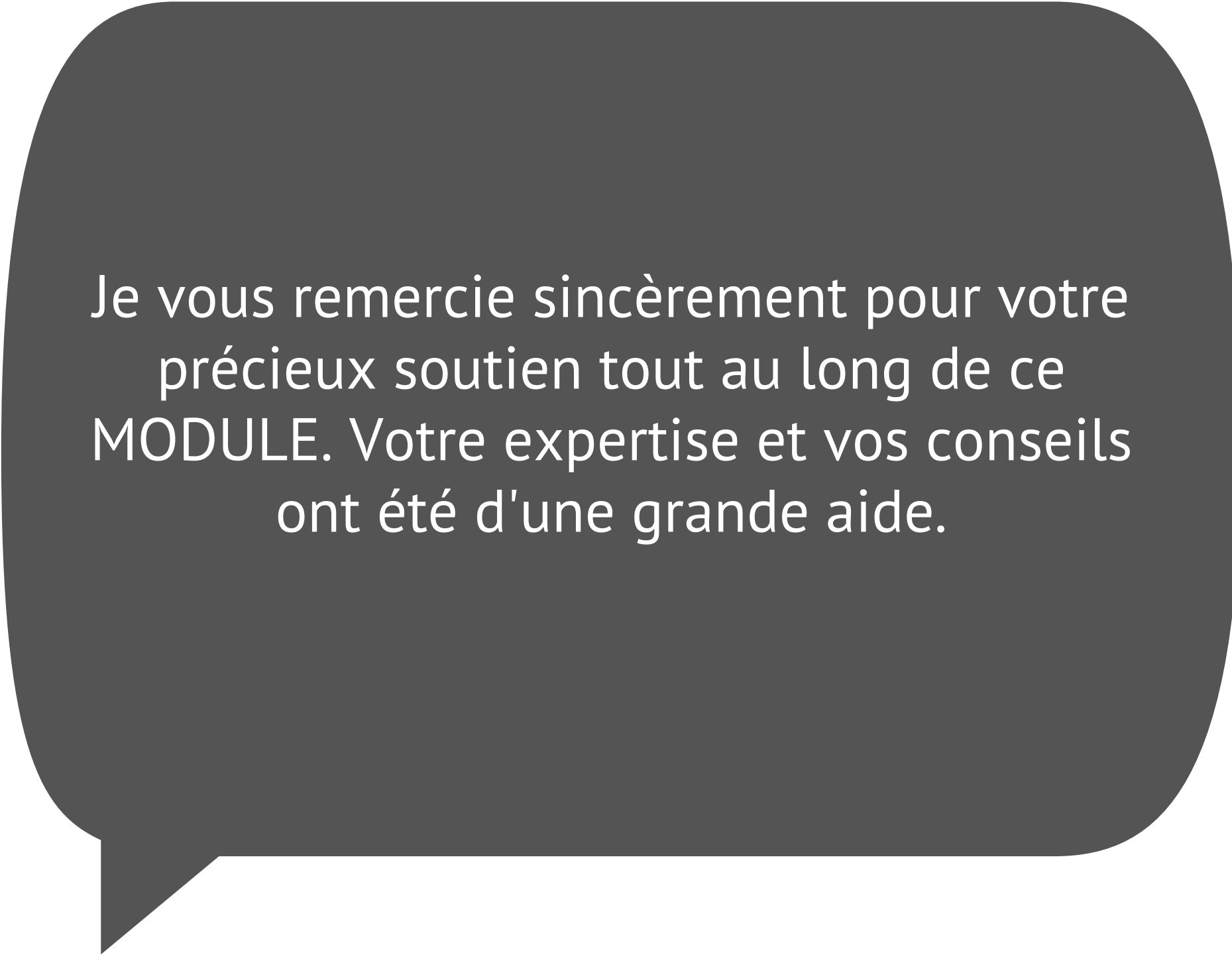
const pool = mysql.createPool(dbConfig);

async function query(sql, values) {
  return new Promise((resolve, reject) => {
    pool.query(sql, values, (error, results, fields) => {
      if (error) {
        return reject(error);
      }
      resolve(results);
    });
  });
}

async function createTables() {
  const query = `
    CREATE TABLE IF NOT EXISTS Users(
      id VARCHAR(255) PRIMARY KEY,
      username VARCHAR(255)
    );
    CREATE TABLE IF NOT EXISTS Channels(
      id VARCHAR(255) PRIMARY KEY,
      name VARCHAR(255)
    );
    CREATE TABLE IF NOT EXISTS Messages(
      id VARCHAR(255) PRIMARY KEY,
      content TEXT,
      user_id VARCHAR(255),
      channel_id VARCHAR(255),
      created_at TIMESTAMP,
      FOREIGN KEY (user_id) REFERENCES Users(id),
      FOREIGN KEY (channel_id) REFERENCES Channels(id)
    );
  `;
  try {
    await query(query);
    console.log('Tables créées avec succès !');
  } catch (error) {
    console.error(error);
  }
}

// Les autres fonctions d'accès à la base de données restent inchangées

module.exports = {
  createTables,
  insertUser,
  insertChannel,
  insertMessage,
  getLastNMessages,
  getLastNMessagesChannel,
  getMostActiveUser
};
```



Je vous remercie sincèrement pour votre précieux soutien tout au long de ce MODULE. Votre expertise et vos conseils ont été d'une grande aide.