**Chatbot System Installation Guide**

**Table of Contents**

1. Prerequisites
2. Installation Steps
   * Setting Up the Environment
   * Installing Dependencies
   * Configuring the Chatbot System
   * Database Setup (MongoDB)
   * API and WebSocket Configuration
   * Running the Chatbot System
3. Post-Installation Tasks
4. Verification and Testing
5. Troubleshooting

**1. Prerequisites**

Before installing the chatbot system, ensure that your environment meets the following requirements:

**1.1 System Requirements**

* **Operating System**: Linux, macOS, or Windows
* **Node.js**: Version 14.x or higher
* **npm**: Version 6.x or higher
* **MongoDB**: Version 4.x or higher
* **Git**: Version control system

**1.2 Network Requirements**

* Stable internet connection for downloading dependencies and accessing external APIs
* Ports **3000** (for the chatbot server) and **8080** (for WebSocket) must be open

**1.3 Software and Tools**

* **Code Editor**: Visual Studio Code, Sublime Text, or any preferred editor
* **Terminal**: Command Line Interface (CLI) for executing commands

**2. Installation Steps**

**2.1 Setting Up the Environment**

1. **Create a Project Directory**:
   * Open your terminal and create a new directory for the chatbot system.

bash

Copy code

mkdir chatbot-system

cd chatbot-system

1. **Clone the Repository**:
   * Clone the chatbot system repository from your version control system (e.g., GitHub).

bash

Copy code

git clone https://github.com/your-repo/chatbot-system.git

cd chatbot-system

**2.2 Installing Dependencies**

1. **Install Node.js Dependencies**:
   * Ensure you are in the project directory. Install all necessary Node.js packages using npm.

bash

Copy code

npm install

* + This command will read the package.json file and install all the required dependencies like Botkit, Express, Faye.js, etc.

1. **Install Global Dependencies (if not already installed)**:
   * **MongoDB**: Install MongoDB from the official [MongoDB download center](https://www.mongodb.com/try/download/community).
   * **PM2** (optional): For running the chatbot as a service.

bash

Copy code

npm install pm2 -g

**2.3 Configuring the Chatbot System**

1. **Create Environment Variables File**:
   * Copy the .env.example file to .env.

bash

Copy code

cp .env.example .env

* + Open the .env file in your code editor and configure the following variables:
    - PORT: The port the chatbot server will run on (default: 3000).
    - MONGO\_URI: The connection string for your MongoDB instance.
    - API\_KEY: Your API key for server-to-server communication.
    - OAUTH\_CLIENT\_ID and OAUTH\_CLIENT\_SECRET: For OAuth authentication.
    - JWT\_SECRET: Secret key for signing JWT tokens.

1. **Botkit Configuration**:
   * Inside the botkit/ directory, configure the bot.js file to include any custom middleware or event listeners specific to your application.

**2.4 Database Setup (MongoDB)**

1. **Start MongoDB**:
   * If MongoDB isn’t running as a service, start it manually:

bash

Copy code

mongod --dbpath /path/to/your/db

1. **Create a Database**:
   * Access the MongoDB shell:

bash

Copy code

mongo

* + Create a new database for the chatbot system:

bash

Copy code

use chatbotDB

1. **Seed Initial Data** (Optional):
   * If you have initial data to seed, run a script or manually insert documents:

bash

Copy code

db.users.insert({ "username": "admin", "password": "yourhashedpassword" })

**2.5 API and WebSocket Configuration**

1. **Configure WebSocket Server**:
   * Inside websocket/, configure the server to handle real-time communication.
   * Ensure the WebSocket server is set to listen on the specified port (default: 8080).
2. **Faye.js Setup**:
   * Install Faye.js as a Node.js package if not already installed.

bash

Copy code

npm install faye

* + Configure Faye.js in your project by setting up the publish-subscribe channels for real-time messaging.

**2.6 Running the Chatbot System**

1. **Start the Server**:
   * Start the chatbot server using Node.js:

bash

Copy code

npm start

* + Alternatively, use PM2 to run the server as a service:

bash

Copy code

pm2 start npm --name "chatbot-system" -- start

1. **Start the WebSocket Server**:
   * In a new terminal window, navigate to the WebSocket directory and start the server:

bash

Copy code

node websocket.js

1. **Start Faye.js Server**:
   * Run the Faye.js server to handle real-time communications:

bash

Copy code

node faye-server.js

**3. Post-Installation Tasks**

**3.1 Testing API Endpoints**

* Use a tool like **Postman** or **cURL** to test your API endpoints and ensure they are functioning correctly.
  + Example:

bash

Copy code

curl -X GET http://localhost:3000/api/status -H "Authorization: Bearer your-oauth-token"

**3.2 Configure the Chatbot Widget**

* Embed the chatbot widget in your web application by including the provided script tag and configuration code.

**3.3 Monitor Logs and Performance**

* Use PM2 or similar tools to monitor server logs, performance, and uptime.

**4. Verification and Testing**

**4.1 Verify Authentication**

* Ensure that all authentication mechanisms (API Keys, OAuth, Username/Password) are working as expected by testing login and access scenarios.

**4.2 Functional Testing**

* Perform end-to-end testing of the chatbot by interacting with it through the widget and verifying that it responds correctly, accesses the database, and retrieves data from external APIs.

**4.3 Real-time Communication Testing**

* Test real-time communication by sending messages and ensuring that WebSocket and Faye.js are correctly handling the real-time updates.

**5. Troubleshooting**

**5.1 Common Issues**

* **Server Not Starting**:
  + Ensure Node.js is correctly installed.
  + Check if the required ports are open and not in use by other applications.
* **Authentication Failures**:
  + Verify that your .env file is correctly configured with the necessary keys and secrets.
  + Ensure OAuth credentials are correctly set up.
* **Database Connection Issues**:
  + Confirm that MongoDB is running and the MONGO\_URI in your .env file is correct.
  + Check if MongoDB authentication is required and set up accordingly.
* **WebSocket Connection Errors**:
  + Verify that the WebSocket server is running and accessible on the specified port.
  + Check for network issues that might be blocking WebSocket traffic.

**5.2 Logs and Debugging**

* **Node.js Logs**: Check the server logs in the terminal or use PM2 to view logs.
* **MongoDB Logs**: Review MongoDB logs for any connection or query issues.
* **WebSocket and Faye.js Logs**: Monitor logs for connection errors or message handling issues.

**Congratulations!** You have successfully installed and configured the Chatbot System. If you encounter any issues or need further assistance, please refer to the Troubleshooting section or contact support.