**Chatbot System User Guide**

**Table of Contents**

1. Introduction
2. System Requirements
3. Getting Started
4. Authentication Methods
   * API Keys
   * OAuth Tokens
   * Username/Password Authentication
5. Using the Chatbot Widget
6. Message Processing and Conversation Flow
7. Real-time Interaction with WebSockets and Faye.js
8. Conversation History and Data Storage
9. Troubleshooting
10. FAQ

**1. Introduction**

Welcome to the **Chatbot System** user guide! This guide will help you get started with using the chatbot, understanding the key features, and leveraging real-time communication. The system is designed to assist you with various tasks by offering an intelligent chatbot interface with secure authentication.

**2. System Requirements**

Before using the chatbot system, ensure you meet the following requirements:

* A modern web browser (Chrome, Firefox, Edge, Safari)
* A valid user account (for OAuth or Username/Password authentication)
* Internet connection for real-time communication

**3. Getting Started**

1. **Access the Chatbot**: Open the website or app that includes the chatbot widget.
2. **Open the Widget**: You should see a chat icon on the bottom right of your screen. Click the icon to open the chatbot widget.
3. **Authentication**: If required, the widget will prompt you to log in. Choose an authentication method:
   * API Key (for developers)
   * OAuth (via token)
   * Username/Password (for standard users)

**4. Authentication Methods**

The chatbot system supports multiple authentication mechanisms to secure access to different features.

**4.1 API Keys**

* **Usage**: API keys are typically used for developers or internal users who need to make server-to-server requests.
* **How to Use**:
  1. In the chatbot widget, select "Developer Mode."
  2. Enter your API key when prompted.
  3. After successful validation, you can start interacting with the bot.

**4.2 OAuth Tokens**

* **Usage**: OAuth tokens provide a secure and standardized way to authenticate users. Tokens are usually generated via a third-party service (e.g., Google, GitHub).
* **How to Use**:
  1. On the login screen, select “OAuth Login.”
  2. You’ll be redirected to the service provider (e.g., Google).
  3. Authorize the chatbot system and you'll receive a token.
  4. The token will be automatically applied, allowing you to interact with the bot.

**4.3 Username/Password Authentication**

* **Usage**: Basic authentication for users who prefer logging in with a simple username and password.
* **How to Use**:
  1. Enter your username and password on the login screen.
  2. After successful login, the chatbot widget will load and be ready for use.

**5. Using the Chatbot Widget**

**5.1 Sending Messages**

* **Step 1**: Type your message in the input box at the bottom of the widget.
* **Step 2**: Press "Enter" or click the "Send" button to submit the message.
* **Step 3**: Wait for the chatbot to process your message and respond.

**5.2 Viewing Responses**

* The chatbot’s response will appear in the widget above the input box in real time. Each response will be context-aware and based on the message you just sent.

**5.3 Understanding Contextual Conversations**

* The chatbot stores the context of each conversation, allowing it to remember previous interactions and provide more relevant responses. You can continue a conversation seamlessly without losing history.

**6. Message Processing and Conversation Flow**

When you send a message, here’s what happens behind the scenes:

1. **Message Sent**: Your message is sent to Botkit, along with your authentication token.
2. **Message Processed**: Botkit analyzes the message to understand its intent and context.
3. **Data Fetched**: If necessary, Botkit will query MongoDB or external APIs to get the data needed to respond (e.g., weather, account info).
4. **Response Sent**: The chatbot sends a response back to you through the widget in real time.

**7. Real-time Interaction with WebSockets and Faye.js**

The system uses **WebSockets** and **Faye.js** for real-time communication.

**7.1 Real-time Updates**

* Every time you send a message or receive a response, WebSockets ensure that the conversation is updated in real-time without the need to refresh the page.

**7.2 WebSocket Connection**

* As soon as you open the chatbot widget, a WebSocket connection is established. This connection allows for fast, bidirectional communication between you and the chatbot.

**7.3 Faye.js Channel**

* Faye.js is responsible for managing the publish-subscribe model, ensuring that all updates and messages are broadcast to the right users in real-time.

**8. Conversation History and Data Storage**

**8.1 Viewing Conversation History**

* The chatbot saves your conversation history in MongoDB. If you refresh the page or log in on a new device, you can continue your conversation from where you left off.

**8.2 Data Security**

* Your data is stored securely using encrypted communication protocols. Authentication tokens (OAuth) and API keys ensure that only authorized users can access sensitive information.

**8.3 Accessing Past Conversations**

* If you need to review a previous conversation:
  1. Click the "Conversation History" button in the widget.
  2. You'll see a list of past conversations with timestamps. Select a conversation to view the full chat.

**9. Troubleshooting**

**9.1 Common Issues**

1. **Unable to Authenticate**:
   * Check if you’ve entered the correct API key or OAuth token.
   * Ensure your username/password is correct.
   * Try refreshing the page or clearing browser cache.
2. **WebSocket Connection Error**:
   * Ensure you have a stable internet connection.
   * If the problem persists, close and reopen the chatbot widget.
3. **Conversation Not Loading**:
   * Your session may have expired. Log in again and check your internet connection.

**10. FAQ**

**Q1: Can I use the chatbot without logging in?**

* **A**: Certain basic features may be available without authentication, but logging in allows access to personalized and secure services.

**Q2: How is my data protected?**

* **A**: The system uses encrypted communication channels, and your data is stored securely in MongoDB. Tokens and API keys ensure that only authorized users can access sensitive data.

**Q3: What happens if my OAuth token expires?**

* **A**: If your OAuth token expires, you’ll be prompted to log in again or refresh your token.

**Q4: Can I integrate this chatbot into my own application?**

* **A**: Yes, developers can use API keys to integrate the chatbot into their own applications using the provided API.

**Thank you for using our Chatbot!**

For further assistance, feel free to contact our support team.

This user guide helps both regular users and developers understand how to interact with the chatbot and resolve common issues while ensuring a smooth experience.