**CHATROOM APPLICATION**

**Documentation: Real-Time Chat-room Application :**

This documentation provides an overview of a real-time chat-room application implemented using Spring Boot for the backend and React for the front end.

**1. Overview:**

The chat-room application allows users to communicate with each other in real time. It consists of two main components:

* **Backend:** Implemented using Spring Boot, it provides WebSocket functionality for real-time messaging.
* **Frontend:** Developed with React, it provides a user interface for users to interact with the chat-room.

**2. Backend Implementation:**

**2.1. Configuring WebSocket:**

* The WebSocket functionality is configured using ‘**WebSocketMessageBrokerConfigurer’** in the ‘**Config’** class.
* It enables WebSocket message handling and sets up message broker destinations.

**2.2. Message Controller:**

* The ‘**MessageController’** class handles incoming messages from clients and sends messages to the message broker.
* It annotates methods with **@MessageMapping** to map specific endpoints and **@SendTo** to specify destinations to send messages.

**2.3. Message Model:**

* The ‘**Message’** class represents the structure of a message sent and received in the chatroom.
* It contains properties such as name (sender) and content (message content).

**3. Frontend Implementation:**

**3.1. HTML Structure:**

* The front end consists of an HTML file (**index.html**) that serves as the main user interface.
* It includes input fields for username and message content, along with buttons for sending messages and logging out.

**3.2. JavaScript Logic:**

* JavaScript code (**script.js**) handles user interactions and WebSocket communication.
* It establishes a WebSocket connection to the backend and handles message sending and receiving.
* The login functionality captures the username entered by the user and initiates the connection to the chatroom.

**4. Running the Application:**

* To run the application, ensure that both the backend (Spring Boot) and frontend (React) servers are started.
* Access the application through a web browser, enter a username to log in, and start sending and receiving messages in the chatroom.

**5. Conclusion:**

The real-time chatroom application provides a simple yet effective platform for users to communicate instantly. It demonstrates the integration of WebSocket technology for real-time communication between clients and a Spring Boot backend serving as the messaging server.