**SkillSwap Messages API Documentation**

**Base URL**

The base URL for all API requests is **https://ws24-skillswap.onrender.com**

**Endpoints**

All endpoints are prefixed with **/api**

**1. Create Message (Users or Admins)**

**Allows users or administrators to create a new message.**

* **Endpoint: /api/messages/create**
* **Method: POST**
* **Access: Users and Admins**
* **Authorization: Required. JWT token must be included in the HTTP-only cookie.**
* **Body Parameters:**
  + **senderId: String (required) - The ID of the sender.**
  + **receiverId: String (required) - The ID of the receiver.**
  + **sentAt: DateTime (required) - The timestamp when the message was sent.**
  + **text: String (required) - The content of the message.**
* **Success Response:**
  + **Code: 201 Created**
  + **Content Example: { "message": "New message successfully created", "new\_message": "messageId" }**
* **Error Response:**
  + **Code: 400 Bad Request**
  + **Content: { "message": "Message could not be created", "error": "Error details" }**

**2. Retrieve Message (Users or Admins)**

**Allows users or administrators to retrieve a specific message by ID.**

* **Endpoint: /api/messages/:id**
* **Method: GET**
* **Access: Users and Admins**
* **Authorization: Required for protected access.**
* **URL Parameters: id - The ID of the message to retrieve.**
* **Success Response:**
  + **Code: 200 OK**
  + **Content Example: { "\_id": "messageId", "senderId": "userId", "receiverId": "userId", "sentAt": "DateTime", "text": "Message content" }**
* **Error Response:**
  + **Code: 404 Not Found**
  + **Content: { "message": "Message not found" }**

**3. Retrieve All Messages (Users or Admins)**

**Allows users or administrators to retrieve all messages in the database.**

* **Endpoint: /api/messages**
* **Method: GET**
* **Access: Users and Admins**
* **Authorization: Required for protected access.**
* **Success Response:**
  + **Code: 200 OK**
  + **Content Example: [{ "message": "Array of all messages objects" }]**
* **Error Response:**
  + **Code: 500 Internal Server Error**
  + **Content: { "message": "An error occurred", "error": "Error details" }**

**4. Retrieve a list of chats for a given user (Users or Admins)**

**This endpoint retrieves a list of chat summaries for the specified user ID. Each chat summary includes the latest message exchanged in the chat, along with the username and possibly the profile image of the chat partner. The chats are sorted by the timestamp of the latest message, with the most recent first.**

* **Endpoint: /api/messages/chatlist/:userId**
* **Method: GET**
* **Access: Users and Admins**
* **Authorization: Required for protected access.**
* **Success Response:**
  + **Code: 200 OK**
  + **Content Example:**

**[**

**{**

**"\_id": "5f3fc1d3c28eca5f5e123456",**

**"username": "john\_doe",**

**"message": {**

**"text": "You: Hello, how are you?",**

**"sentAt": "2024-04-12T10:42:11.958Z"**

**}**

**},**

**{**

**"\_id": "5f3fc1d3c28eca5f5e654321",**

**"username": "jane\_smith",**

**"message": {**

**"text": "Looking forward to our meeting.",**

**"sentAt": "2024-04-11T08:15:30.123Z"**

**}**

**}**

**]**

* **Error Response:**
  + **Code: 500 Internal Server Error**
  + **Content: { "message": "An error occurred", "error": "Error details" }**

**5. Retrieve All Messages between 2 Users (Users or Admins)**

**This endpoint fetches all messages exchanged between two specific users, regardless of who was the sender or the receiver. Messages are returned in the order they were received by the database, which typically aligns with the chronological order of sending.**

* **Endpoint: /api/messages/:senderId/:receiverId**
* **Method: GET**
* **Access: Users and Admins**
* **Authorization: Required for protected access.**
* **Path Parameters:**
  + **senderId (required): The user ID of one of the users in the chat.**
  + **receiverId (required): The user ID of the other user in the chat.**
* **Success Response:**
  + **Code: 200 OK**
  + **Content Example:**

**[**

**{**

**"\_id": "60afcefe1234abc123456789",**

**"text": "Hey, how's it going?",**

**"createdAt": "2024-04-12T10:00:00Z",**

**"senderId": {**

**"\_id": "5f3fc1d3c28eca5f5e123456",**

**"username": "john\_doe"**

**},**

**"receiverId": "5f3fc1d3c28eca5f5e654321"**

**},**

**{**

**"\_id": "60afcefe1234abc123456790",**

**"text": "Great, thanks! And you?",**

**"createdAt": "2024-04-12T10:01:30Z",**

**"senderId": {**

**"\_id": "5f3fc1d3c28eca5f5e654321",**

**"username": "jane\_smith"**

**},**

**"receiverId": "5f3fc1d3c28eca5f5e123456"**

**}**

**]**

* **Error Response:**
  + **Code: 500 Internal Server Error**
  + **Content: { "message": "An error occurred", "error": "Error details" }**

**6. Update Message (Users or Admins)**

**Allows administrators to update information for a specific message by ID.**

* **Endpoint: /api/messages/update/:id**
* **Method: PUT**
* **Access: Admins only**
* **Authorization: Required. JWT token must be included in the HTTP-only cookie.**
* **URL Parameters: id - The ID of the message to update.**
* **Body Parameters: The fields to update (e.g., text: "Updated message content").**
* **Success Response:**
  + **Code: 200 OK**
  + **Content Example: { "message": "Update successful", "updated\_message": { updated message object } }**
* **Error Response:**
  + **Code: 404 Not Found**
  + **Content: { "message": "Message not found" }**

**7. Delete Message (Admin Only)**

**Allows administrators to delete a specific message by ID.**

* **Endpoint: /api/messages/delete/:id**
* **Method: DELETE**
* **Access: Admins only**
* **Authorization: Required. JWT token must be included in the HTTP-only cookie.**
* **URL Parameters: id - The ID of the message to delete.**
* **Success Response:**
  + **Code: 201 Created**
  + **Content Example: { "message": "Message successfully deleted", "message": { deleted message object } }**
* **Error Response:**
  + **Code: 400 Bad Request**
  + **Content: { "message": "An error occurred", "error": "Error details" }**

**Notes**

* **Secure Connections:** It is highly recommended to use HTTPS for all API interactions to ensure data privacy and security.
* **Authentication:** All protected routes require authentication. A valid JWT token must be presented in the request's HTTP-only cookie.
* **Roles and Permissions:** Access to certain actions is restricted based on the user's role, as indicated in the JWT token.