

#	Endpoint	Description	Request Body	Response
1	POST /api/chat/send	Send a chat message to a user or a group.	<pre>json { "to": ["username1", "username2"], "group": "group_id", "payload": "Hello, how are you?", "attachments": [{ "type": "image", "url": "https://example.com/image.jpg" }] }</pre>	<pre>json { "message": "Chat sent successfully." }</pre>
2	POST /api/chat/reply	Reply to an existing chat message.	<pre>json { "chatId": "chat_id", "payload": "I'm doing well!", "attachments": [{ "type": "video", "url": "https://example.com/video.mp4" }] }</pre>	<pre>json { "message": "Reply sent successfully." }</pre>
3	POST /api/chat/create-group	Create a new group.	<pre>json { "name": "My Group", "participants": ["username1", "username2"], "about": "A group for discussing various topics." }</pre>	<pre>json { "message": "Group created successfully." }</pre>
4	POST /api/chat/add-participants/:groupId	Add participants to an existing group.	<pre>json { "participants": ["username3", "username4"] }</pre>	<pre>json { "message": "Participants added successfully." }</pre>
5	PUT /api/chat/leave/group/:groupId	Leave a group.	None	<pre>json { "message": "Left group successfully." }</pre>
6	PUT /api/chat/make-admin/:groupId	Make a participant an admin in a group.	<pre>json { "participantUsername": "username5" }</pre>	<pre>json { "message": "Participant made an admin successfully." }</pre>
7	GET /api/chat/group/:groupId	Retrieve details about a group.	None	<pre>json { "group": { "name": "My Group", ... } }</pre>

8	GET /api/chat/user-groups	Retrieve groups where the user is a participant or admin.	None	json { "groups": [{ "_id": "group_id", "name": "My Group", ... }] }
9	PUT /api/chat/update-group/:groupId	Update group details.	json { "name": "New Group Name", "profilePicture": " https://example.com/new_image.jpg ", "about": "Updated group description." }	json { "message": "Group details updated successfully." }
10	GET /api/chat/group/:groupId/history	Retrieve group history.	None	json { "history": [{ "user": "user_id", "username": "username", "action": "Group name changed to New Group Name", "date": "2023-12-01T12:00:00.000Z" }] }
11	GET /api/chat/group/:groupId/participants	Retrieve group participants.	None	json { "participants": [{ "_id": "user_id", "username": "username", "username": "username", ... }] }
12	PUT /api/chat/force-leave/group/:groupId/:participantId	Force a participant to leave a group.	None	json { "message": "Forced participant to leave group successfully." }
13	PATCH /api/chat/mark-as-read/:chatId	Mark a chat message as read.	None	json { "message": "Message marked as read", "chat": { ... } }

14	DELETE /api/chat/delete-all/:userId	Soft delete all messages for a specific user.	None	json { "message": "All messages deleted", "result": { ... } }
15	POST /api/chat/archive/:userId	Archive messages for a specific user.	None	json { "message": "Messages archived", "result": { ... } }
16	POST /api/chat/restore-archived/:userId	Restore archived messages for a specific user.	None	json { "message": "Archived messages restored", "result": { ... } }
17	POST /api/chat/react/:chatId	Add a reaction to a chat message.	json { "emoji": "👍" }	json { "message": "Reaction added successfully", "chat": { ... } }
18	PUT /api/chat/edit/:chatId	Update or edit a message within ten minutes of being sent.	json { "content": "Edited message content" }	json { "message": "Message updated successfully", "chat": { ... } }
19	DELETE /api/chat/delete/:chatId	Soft delete a chat message.	None	json { "message": "Message deleted successfully", "chat": { ... } }
20	GET /api/chat/search	Search messages based on a query string.	Query Parameter: query (string)	json { "matchedChats": [{ ... }] }
21	PATCH /api/chat/receive/:chatId	Mark a chat as received by the authenticated user.	None	json { "message": "Chat marked as received", "chat": { ... } }
22	GET /api/chat/unread	Retrieve unread messages for the authenticated user.	None	json { "unreadMessages": [{ ... }] }

23	GET /api/chat/not-received	Retrieve messages not received by the authenticated user.	None	json { "notReceivedMessages" : [{ ... }] }
24	GET /api/chat/by-attachment/:attachmentType?	Retrieve messages by attachment type.	URL Parameter: attachmentType (string, optional)	json { "messagesByAttachmentType": [{ ... }] }
25	GET /api/chat/	Retrieve chats where the user is either the sender or receiver.	None	json { "userChats": [{ ... }] }
26	DELETE /api/chat/groups/:groupId	Delete a group and its accompanying chats.	URL Parameter: groupId (string)	json { "message": "Group and chats deleted successfully", "deletedChats": { ... } }