

QuickTalk API

Overview

This comprises of backend APIs that allows user registration, private and group chat room creation and addition of users to group chat rooms. This API also enables to fetch list of all registered users and room details of each user.

Version information

Version : 1.0

URI scheme

Host : localhost:8080

BasePath : /

Tags

- Room : Room Controller
- Users : User Controller

Paths

Add users to a group chat room with the specified group room ID

POST /group/users

Description

This API is used to add a single or list of user IDs to an existing group room ID. Returns the group room ID and message along with 200 status code on success, 404 status code if user addition to group room fails and 400 status code if request validation fails.

Parameters

Type	Name	Description	Schema
Body	addUserToGroupRequest <i>required</i>	addUserToGroupRequest	AddUsersToGroupRoomRequestBean

Responses

HTTP Code	Description	Schema
200	OK - {"roomId": "1", "message": "Successfully added userIds [4, 5] to groupRoomId 1"}	RoomResponseBean
201	Created	No Content
400	Bad Request - {"roomId": null, "message": "User Id List cannot be empty"}	RoomResponseBean
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found - {"roomId": null, "message": "Failed to add users to group room Id"}	RoomResponseBean

Consumes

- `application/json`

Produces

- `*/*`
- `application/json`

Tags

- Room

Create or fetch a group chat room by user IDs

POST /rooms/group

Description

This API creates a new group chat room for the specified user IDs if it does not already exist, or fetches the existing room if it does. Returns a group room ID and message along with 200 status code on success, 404 status code if room creation/fetching fails and 400 status code if request validation fails.

Parameters

Type	Name	Description	Schema
Body	groupRoomRequest <i>required</i>	groupRoomRequest	GroupRoomRequest Bean

Responses

HTTP Code	Description	Schema
200	OK - {"roomId": "1", "message": "null"}	RoomResponseBean
201	Created	No Content
400	Bad Request - {"roomId": null, "message": "User Id List cannot be empty"}	RoomResponseBean
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found - {"roomId": null, "message": "Failed to get or create group room"}	RoomResponseBean

Consumes

- `application/json`

Produces

- `*/*`
- `application/json`

Tags

- Room

Create or fetch a private chat room by user IDs

POST /rooms/private

Description

This API creates a new private chat room for the specified user IDs if it does not already exist, or fetches the existing room if it does. Returns a private room ID and message along with 200 status code on success, 404 status code if room creation/fetching fails and 400 status code if request validation fails.

Parameters

Type	Name	Description	Schema
Body	privateRoomRequest <i>required</i>	privateRoomRequest	PrivateRoomRequest Bean

Responses

HTTP Code	Description	Schema
200	OK - {"roomId": "1", "message": "null"}	RoomResponseBean
201	Created	No Content
400	Bad Request - {"roomId": null, "message": "First UserId cannot be empty"}	RoomResponseBean
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found - {"roomId": null, "message": "Failed to get or create private room"}	RoomResponseBean

Consumes

- `application/json`

Produces

- `*/*`
- `application/json`

Tags

- Room

Fetches all rooms by user ID

```
GET /rooms/user/{userId}
```

Description

This API returns list of all rooms of the requested user ID if found, or a 404 status code if no rooms found.

Parameters

Type	Name	Description	Schema
Path	userId <i>required</i>	ID value for the user whose rooms details you need to retrieve	integer (int32)

Responses

HTTP Code	Description	Schema
200	Success	RoomProjection
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- **/**
- *application/json*

Tags

- Room

Registers a new user

POST /user

Description

This API creates a new user with provided details. The request body must include the user's name, email, and password. Returns userId along with 201 status code on success, 404 status code if registration fails and 400 status code if request validation fails along with message.

Parameters

Type	Name	Description	Schema
Body	registerUserRequest <i>required</i>	registerUserRequest	RegisterUserRequest Bean

Responses

HTTP Code	Description	Schema
200	OK - {"userId": "1", "message": "Successfully registered johnDoe"}	MessageResponse Bean
201	Created	No Content
400	Bad Request - {"userId": null, "message": "Password cannot be empty"}	MessageResponse Bean
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found - {"userId": null, "message": "Failed to register user johnDoe"}	MessageResponse Bean

Consumes

- `application/json`

Produces

- `*/*`
- `application/json`

Tags

- Users

Fetches all registered users

GET /users

Description

This API returns list of all registered users if found, or a 404 status code if no user is found.

Responses

HTTP Code	Description	Schema
200	Success	UserProjection
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Produces

- */*
- application/json

Tags

- Users

Definitions

AddUsersToGroupRoomRequestBean

Request model for adding users to a group chat room

Name	Description	Schema
groupRoomId <i>required</i>	Room ID of an existing group Example : 1	integer (int64)
userIdList <i>required</i>	List of one or more user IDs to be added in an existing group Example : [1, 2, 3]	< integer (int32) > array

GroupRoomRequestBean

Request model for creating/fetching group chat room

Name	Description	Schema
groupRoomName <i>required</i>	Indicates name of the group room Example : "John-Jane-Richard Group Chat"	string
userIdList <i>required</i>	List of user IDs for the group room Example : [1, 2, 3, 4]	< integer (int32) > array

MessageResponseBean

Represents user registration ID & corresponding message

Name	Description	Schema
message <i>required</i>	Indicates message regarding user registration request Example : "Successfully registered John Doe"	string
userId <i>optional</i>	Indicates user ID of the registered user. Can be null if registration fails Example : "1"	string

PrivateRoomRequestBean

Request model for creating/fetching private chat room

Name	Description	Schema
firstUserId <i>required</i>	Details of first user ID of private chat Example : "1"	string
secondUserId <i>required</i>	Details of second user ID of private chat Example : "2"	string

RegisterUserRequestBean

Request model for user registration

Name	Description	Schema
email <i>required</i>	Indicates email ID of the user Example : "john@abc.com"	string
password <i>required</i>	Indicates user password Example : "qwerty123"	string
username <i>required</i>	Indicates name of the user Example : "johnDoe"	string

RoomProjection

Name	Description	Schema
roomId <i>required</i>	Unique identifier of the room Example : 1	integer (int64)
roomName <i>required</i>	Indicates name of the room Example : "John-Jane-Richard Group Chat"	string
roomType <i>required</i>	Indicates type of room. Possible values: single or group Example : "group"	enum (single, group)

RoomResponseBean

Name	Description	Schema
message <i>optional</i>	Indicates message regarding chat room request. Example : "Successfully added userIds [4, 5] to groupRoomId 1"	string
roomId <i>optional</i>	Indicates room ID. Can be null if request fails Example : "1"	string

UserProjection

Name	Description	Schema
email <i>required</i>	Indicates email ID of the user Example : "john@abc.com"	string
userId <i>required</i>	Unique identifier of the user Example : 1	integer (int32)
username <i>required</i>	Indicates name of the user Example : "johnDoe"	string