Contents:

Task 1: RESTful API server

Implement a RESTful API for a chat server. The server API will contain at least these resources and operations:

Users:

Route: /api/users
Operations: Get all, get one, add one (e.g. register), delete one (e.g. deregister)

Chat-Rooms:

Route: /api/rooms
Operations: Get all, get one, add one.

Room users:

Route: /api/room/<room-id>/users
Operations: Get all, add one(e.g. join)
Restrictions: Only registered users can join

Messages:

Route: /api/room/<room-id>/messages
Operations: Get all
Restrictions: Only users in the room can get messages.

Route: /api/room/<room-id>/<user-id>/messages
Operations: Get all, add one
Restrictions:

- Only users in the room can get or add messages. Only registered user-id's should be permitted as <user-id>
- All requests except the one adding a new user, will require userID to be passed in with the request as JSON.

Task 2: Clients and bots.

1. Implement a "client program" or library, that can connect to server described above and perform all of the supported operations.
2. Implement at least 4 "chat bots".
    - Register as a user once
    - Join an existing room and create a new one
    - Post several messages in all the roms it's a part of and also fetch all messages in those rooms. Great bots respond to other messages. Lesser bots just post things. Both are welcome.
    - Implement a basic user interface for a human to observe and optionally interact with the chat rooms. The user interface can be a simple terminal program or a web page that lets the user select which of the actions accessible in the API it wants to take (for example, fetch all messages from room 42) and then asks for input to that action where applicable (for example, if the action is to post a message to room 42, allow the user to type in the message they want to send)