

Exercise 1 – 10 Points

At <http://socket.io/get-started/chat/> you can find a simple chat application developed with node.js. You can use this source code as a basis for your own development.

Enhance the socket.io chat room server that

- a new user has to register with the website first by providing a chat name
- in the chat room, all messages should be displayed in chronological order with a timestamp and the chat name of the user who has posted the message.
- a marked message is generated and sent to the chat room each time a new user connects to the chat room or leaves the chat room.
- a user can request a list of all online users using the “\list” command; that list is only sent to the requestor.
- a user can send a message to one dedicated recipient (private message) instead of to all online users in the chat room
- a user can send multi-media files to the participants of a chat room or via a private message, e.g., pictures, movies, sound files.

Please note, that the source code developed in this exercise will be used in further exercises. Hence, develop with maintainability and extensibility in mind!

Submission:

- Prepare design documentation of your solution. It should at least contain: use case diagrams, component diagram, design mockups
- Include at least one lesson-learned item about node.js programming in your document. This can be of any domain, e.g., architecture, coding, or behavioral.
- Implement your solution and prepare a demonstration of your solution
- Upload the design documentation and source code (as one zip file) to RELAX. All uploaded files have to contain the name and student ID of all members of your group!
 - Do not include any node-modules in your source code but rather define dependencies in package.json!

Note that any attempt of cheating, i.e., copying, will result in zero points in the hands-on part of the lecture.