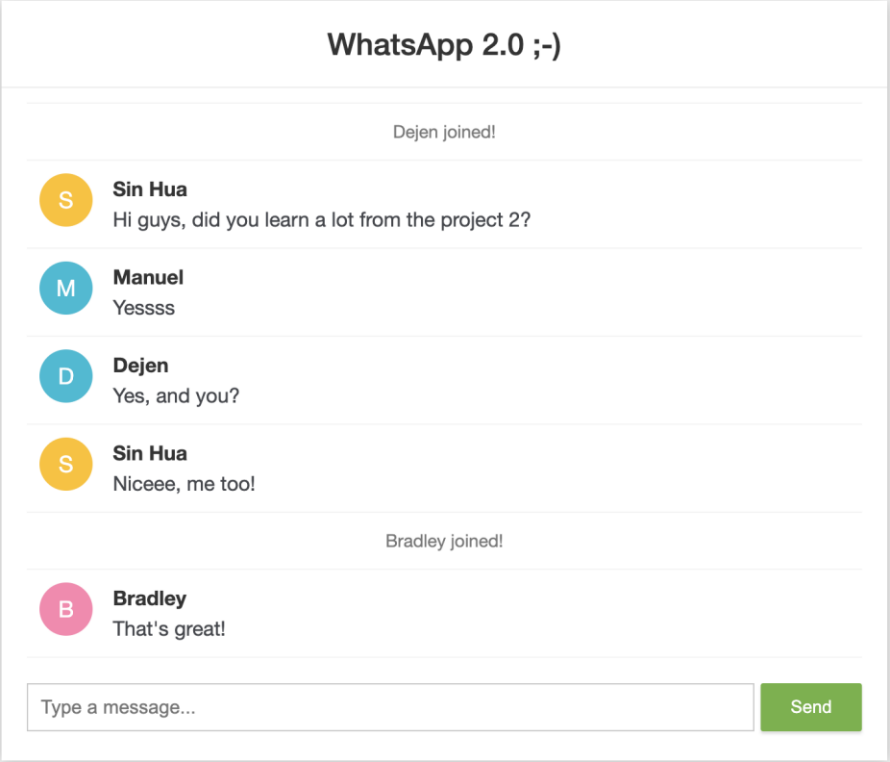
Fachhochschule Nordwestschweiz

Project Documentation: ChatServer

- Our own approach



Manuel Buser, Sin Hua Chai & Dejen Teklit

26.11.2023

FHNW School of Business, Olten

Bradley Richards

Business Information Technology

Software Engineering Leadership SWEL

Type of Project

We first wanted to extend the client. Dejen started the Project on following github repository using JavaFX:

https://github.com/dejenteklit/chatroom-client-server

We also tried to implement our own server on following repository:

https://github.com/manuu1999/ChatProjectServer.git

But then we had the idea to use springboot and made again a new repository which is the current repository. It was really hard for us to decide which way to go because there were to many possibilities and each of them had its own disadvantages and advantages.

Furthermore, we used WebSocket instead of HTTP Connection to make the chat application more dynamic, since with the Websocket API you can leave the connection open.

With the integration of Spring Boot, the WebSocket communication is facilitated by annotations and configurations provided by the Spring Framework. WebSocket functionality is embedded within the application itself, removing the necessity for a separate server class to handle socket connections explicitly.

We also tried to implement the H2 Database to create the new features for login and register - However, we did not fully manage to run it However, the whole Structure and the Code are finished and visible under the new Branch called "NewFeatures" We hope that even though they are not completely working yet they will also be considered for the grading. Thanks :-)

But for this project we decided to go with the working code on the master branch – also the structure below and the class diagram is based on the code from the master branch.

Structure of the Code:

This is a Spring Boot application for a simple chat platform using WebSocket for real-time communication. Here is the structure and functionality of the project:

Java Packages:

com.example.ChatProject_SWEL_SpringBoot.chat:

- Contains the ChatController handling WebSocket message mappings for sending messages and adding users to the chat.
- Defines ChatMessage representing the message structure.
- Includes MessageType, an enum specifying different message types like CHAT, JOIN, and LEAVE.

com.example.ChatProject_SWEL_SpringBoot.confic:

- WebSocketConfig sets up the WebSocket configuration, defining endpoints and message brokers.
- WebSocketEventListener listens for WebSocket events like user disconnects or joins and sends notifications.

com.example.ChatProject_SWEL_SpringBoot:

- The main application class ChatProjectSwelSpringBootApplication initializes the Spring Boot application.

HTML/CSS/JS:

HTML (index.html):

- Two main sections: username-page and chat-page.
- Users input their username on the username page to enter the chat.
- The chat interface includes a message area and a form to send messages.
- Utilizes SockJS and Stomp for WebSocket communication.
- Links to CSS and JS files for styling and functionality.

CSS (main.css):

- Contains styles for different elements of the chat application, including the username page, chat container, message display, and form controls.

JavaScript (main.js):

- Manages the WebSocket connection.
- Handles user connection, sending and receiving messages, and updating the chat interface based on events (e.g., user join, leave, or send a message).

Overall Structure:

Server-Side (Java):

- Manages WebSocket connections and message handling.
- Defines message types and controllers for sending/receiving messages and user actions.

Client-Side (HTML/CSS/JS):

- Provides a user interface for entering usernames and chatting.
- Establishes WebSocket connections, sends messages, and displays incoming messages.
- The code seems to create a straightforward chat application with basic functionalities for sending messages, joining, and leaving the chatroom.

Class Diagram

The class diagram is also on our github Project as a jpg file.