

WebSocket-test.html

Dette dokumentet viser hvordan man benytter en WebSocket-klient for testing av backend-kommunikasjon via SockJS og STOMP. Målet er å demonstrere tilkobling, meldingsoverføring og feilsøking på en enkel og oversiktlig måte. – Dette er basisen for implementasjonen av chat-funksjonaliteten i prosjektet.

Teknologi og Oppsett

Filen er basert på HTML og JavaScript. I koden benyttes SockJS for å etablere tilkoblingen og STOMP.js for å implementere meldingsprotokollen. Brukeren angir en server-URL (for eksempel <http://localhost:8080/ws>) og sender deretter meldinger til backend. Grensesnittet består av et felt for URL, inndatafelter for sender og mottaker, samt et tekstområde for meldingen. Alt logges i et eget område for å gjøre det enklere å feilsøke og verifisere kommunikasjonen.

Funksjonalitet

Når brukeren trykker på «Connect», opprettes en WebSocket-tilkobling. Tilkoblingsstatus vises på skjermen med korte tilbakemeldinger som «Connecting» og «Connected». Dersom noe går galt, logges en kort feilmelding. Når tilkoblingen er etablert, kan brukeren sende en melding ved å fylle ut sender-ID, mottaker-ID og meldingstekst. Meldingen sendes til backend via destinasjonen `/app/chat.sendMessage`. Klienten mottar svar i emnet `/topic/messages`, samt oppdateringer på `/topic/messages.read` og `/topic/messages.update`. I tillegg finnes en funksjon for å sende en enkel "ping", som tester at tilkoblingen er stabil.

Testprosedyre

For å teste løsningen må backend-serveren være i gang og korrekt konfigurert for WebSocket-kommunikasjon. Studenten åpner `websocket-test.html` i en moderne nettleser, fyller ut URL og klikker på «Connect». Deretter sendes testmeldinger for å verifisere at meldingsutvekslingen fungerer. Til slutt testes frakobling med knappen «Disconnect», som også oppdaterer visuell tilbakemelding.