

Proiect Web - ChatTogether

Balint Cristian

January 15, 2024

Contents

1	Introducere	2
2	Încadrare în context	2
3	Motivația tehnică	3
3.1	SpringBoot Framework	3
3.2	React Framework	3
4	Initializare proiecte de dezvoltare a aplicației	4
4.1	Proiectul java pentru backend	4
4.2	Proiectul JavaScript pentru frontend	5
5	Implementare	5
5.1	BackEnd	5
5.2	Front End	7
6	Mentiiuni	8
7	Bibliografie	9

1 Introducere

În contextul unor noi tehnologii aparute de la o zi la alta este trivial sa ne folosim de ele pentru dezvoltarea aplicatiilor fie ele web sau nu. Astfel acest proiect foloseste doua framework-uri populare in dezvoltarea aplicatiilor web: SpringBood si React.

2 Încadrare în context

Proiectul consta in crearea unei aplicatii de chatting folosind SpringBoot si alte unelte sau concepte precum WebSockets sau dependency injection in cazul mediului java. In cadrul mediului javascript s-a utilizat react, typescript si scss. React are la baza scrierea de cod functional si de componente functionale. SpringBoot respectand metoda clasica de a scrie vederi, modele si controllere.

3 Motivația tehnică

Alegerea frameworkului springboot a fost o decizie luata de profesorul responsabil cu disciplina web. Iar alegerea frameworkului react a fost o alegere personala intrucat un istoric cu orice tehnologie ajuta la dezvoltarea unei aplicatii.

3.1 SpringBoot Framework

Avantajele și dezavantajele SpringBoot:

- + Conceptele de baza sunt usor de inteles.
- + Are o comunitate mare si multa documentatie pentru diverse probleme.
- + Este bun la configurarea serverului web si a endpointurilor sale.
- Daca nu ai lucrat cu mvc poate fi unpic ciudat pana te acomodezi.
- ? Are un mod anume de a scrie codul si este restrictiv in anumite aspecte.
In functie de ce vrem sa obtinem poate fi un avantaj sau dezavantaj.
Pe scurt este un framework foarte util daca stim ce vrem sa facem cu el.

3.2 React Framework

Avantajele și dezavantajele React:

- + Are o comunitate mare si multa documentatie pentru diverse probleme.
- + Este bun la dezvoltarea frontendului si crearea unor componente functionale in acest aspect
- + Are la baza conceptul de componenta si stare a componentei(sau a aplicatiei)
- Are la baza conceptul de hooks, care desi are rolul de a simplifica codul prin a exclude clasele si metodele pentru crearea si gestionarii starilor unor componente, este unpic mai greu de inteles
- Are dependinte inechitate cea ce cauzeaza probleme de compatibilitate care ingreuneaza procesul de dezvoltare web daca nu este instalat corect.
- ? Nu respecta mvc, poate fi vazut ca upside sau downside in functie de preferinte si experienta.

4 Initializare proiecte de dezvoltare a aplicatiei

4.1 Proiectul java pentru backend

Proiectul foloseste maven pentru Gestionarea Proiectului Software respectiv in-telegera structurii proiectului software, adaugarea si instalarea de dependinte si alte lucruri specifice frameworkului. Dependintele unui proiect pot fi adaugate si instalate manual din fisier-ul .pom(Project Object Model) al proiectului.

Insa noi am utilizat un tool furnizat de spring boot, respectiv Spring initial-izaer. Acesta se ocupa cu initiliazarea proiectelor, adaugarea unor dependinte de baza specifice sprin boot, selectarea versiunii de springboot, selectara celui "project manager" care in cazul nostru este maven si a limbajului in care se dezvolta(java, kotlin sau groovy).

Dependintele alese din spring initializer:

Spring Web WEB

Dezvoltare web, incluzand si conceptul rest, utilizand Spring MVC. Utilizeaza implicit Apache Tomcat ca spring boot server care poate gestiona multiple apli-catii web.

WebSocket MESSAGING

WebSocket configurare si implementare websocket cu ajutorul SockJS si STOMP.

Spring Boot DevTools DEVELOPER TOOLS

Furnineaza utilitati in procesul de dezvoltare precum aplicattion fast restart, live reload si configuratii pentru a usura dezvoltarea proiectului.

Spring Session WEB

Furnizeaza un API si implementari pentru gestionara informatiilor din cadrul Sesiunii Utilizator.(User Session)

Spring Security SECURITY

Framework de securitate personalizabil pentru gestionarea accesului si controlu-lui asupra aplicatiei.

Spring Data JPA SQL

Persistenta datelor in sql stores folosind java persistence api.

MySQL Driver SQL

Contine driverul java dabase connector pentru mysql.

4.2 Proiectul JavaScript pentru frontend

Proiectul foloseste yarn pentru gestionarea proiectului software respectiv intelegerea structurii proiectului software, adaugarea si instalarea de dependinte si alte lucruri specifice frameworkului. Proiectele react se pot initiliza dintr-o singura companda si se pot pasa anumite templateuri prestabilite, insa pentru adaugarea dependintelor utilizam: `yarn add DependencyName`.

Noi am preferat se folosim vite ca server de dezvoltare web, acesta vine implicit cu lucruri similare dependintei "Develeper Tools de pe backend", aceeaasi idee de hot module replacement. Un restart partial al serverului pentru a incarca modificarile efectuate.

Comanda pentru initializarea proiectului cu vite:
`yarn create vite my-react-app --template react-ts`

Parametrul optional de template este react-ts, acesta presupune un proiect react cu typescript. Insa puteam utiliza si zvelte sau vue, cu typescript sau fara. Pe pagina celor de la vite sunt toate templateurile mentionate.

Dependintele alese:

Scss

Scss un limbaj de scriptare preprocessat care este interpretat sau compilat in Cascading Style Sheets (CSS).

Typescript

Este un superset pentru javascript, aceste cu tipuri predefine si restrictii de tip. Acesta aduce si interfete sau alte lucruri pentru a restrictiona si preveni erorile de tip specifice javascript.

React-router-dom

Dependita pentru routing web prin manipularea dom-ului.

SocketJS-Client

Creeaza un obiect WebSocket pentru client.

StompJs

Permite conectarea la un STOMP broker de pe un WebSocket.

5 Implementare

5.1 BackEnd

O sa construim un backend care are rolul atat de furniza date similiar unui api, cat si de a furniza un login extern front-endului care are si el un server. Prima

oara adnotam clasa principala care contine metoda main cu `@SpringBootApplication`.

Punand ulterior in metoda main linia de cod: `SpringApplication.run(NumeleClasei.class, args)`; O sa utilizam la nivel de proiect 7 pachete aditionale clasei Application cu metoda main.

Pachete sunt:

1. Security - contine configurarea si serviciul de securitate
 - CustomUserDetailsService
 - Security Config
2. Model - contine modelarea datelor sub forma de entitati care vor fi transferate sub forma unor obiecte.
 - ConversationDTO
 - ConversationUserDTO
 - MessageDTO
 - UserCredentialsDTO
 - UserDataDTO
3. Persistence - clase de descriere a tabelelor din baza de date
 - Conversation
 - Conversation
 - Message
 - User
4. Repository - contine metodele de interogare a bazei de date utilizate ulterior de servicii
 - ConversationRepository
 - ConversationUserRepository
 - MessageRepository
 - UserRepository
5. Controller - contine controlerele care se ocupa cu Transferul Stariilor Reprezentative(REST) pe diverse endpointuri, returand de obicei in cazul metodelor de tip get apelarea unei metode a unui serviciu, care la randul ei returneaza un DTO.
 - AuthController
 - ConversationController
 - ConversationUserController
 - MessageController

- UserController
- 6. Service - contine serviciile specifice entitatilor care returneaza obiecte de transfer al datelor
 - ConversationService
 - ConversationUserService
 - MessageService
 - UserService
- 7. WebSocket - contine configurarea websocketului
 - WebSocketConfig
 - SocketController

5.2 Front End

In cadrul frontend-ului avem un single page application(SPA). Fisierul main.tsx contine crearea radacinii dom-ului react fiind unul dintre cei mai importanti pasi in utilizarea react. In App.tsx am utilizat AuthProvider-ul si in functie de starea de login a utilizatorului acesta, are sau nu acces la pagina Home de la ruta /home. Daca nu este logat, acesta este redirectionat automat la login in incercarea de accesa ruta respectiva. Odata autentificat userul este redirectionat la /home si acolo are interfata de chatting. Aceasta vine cu un Header care contine lista conversatiilor user-ului, nickname-ul userului si un button de logout. Sub Header se afla container-ul principal care isi actualizeaza aspectul in functie de date. Odata logati, la randarea componentei Home se creeaza conectiunea la web socket si se obtin datele utilizatorului printr-un fetch, acestea fiind ulterior puse in header.

6 Mentiuni

Nu stiu ce sa mai zic, cam atat. Am intampinat o eroare pe backend care nu am stiut cum sa o rezolv si am facut atatea schimbari ca nici logout-ul nu mai functioneaza corect. Eroarea tine de crearea tabelelor si respectiv un id care nu se creeaza ca auto increment primary key, indiferent daca am setat indenty pe el. Iar cu socketul, nu am reusit sa-l configurez pentru ca nu merge salvarea mesajelor in baza de date, pentru ca primesc o eroare referitoare la lipsa unui id default, deci problema anterioara. Asa ca am ramas cu ce am pana acum. Cat despre commit-uri nu intilizasem un github repository pana acum, ca am uitat. Am vazut abordari pentru crearea unui SocketController fara un MessageDTO si fara o baza de date, dar ce sens are sa fur codul respectiv daca nu ma folosesc de ce am. Decat sa am o chesti falsa din punct de vedere vizual, am zis ca nu iau nimic.

7 Bibliografie

References