



Web sockets and security

Student: Bode Andreea-Nicoleta

Profesor: Chis David

Facultatea de Automatica si Calculatoare

Specializarea: Tehnologia Informatiei

Cuprins

1. Cerinta.....	3
2. Implementare.....	4
3. Read Me.....	9
Frontend	9
Device, User Microservices, Producer si Consumer, Chat.....	10
3. Bibliografie.....	10

1. Cerinta

Develop a chat microservice and an authorization component for the Energy Management System. The authorization component should provide secured access of users to systems' microservices. The chat microservice should allow communication between the users and the administrator of the system, allowing them to ask questions and receive answers

Frontend

- Interfața grafică a utilizatorului (UI) care permite autentificarea utilizatorilor, gestionarea conturilor și dispozitivelor inteligente, vizionarea graficelor și a notificării.
- Interfața grafică permite comunicarea între 2 utilizatori: unul de tipul Admin și altul de tipul User

Microserviciu pentru Managementul Utilizatorilor

- Gestionarea conturilor de utilizator: adăugare, citire, actualizare și ștergere (CRUD).
- Autentificarea și autorizarea utilizatorilor.
- Crearea și gestionarea mapărilor dintre utilizatori și dispozitive.

Microserviciu pentru Managementul Dispozitivelor

- Gestionarea dispozitivelor inteligente: adăugare, citire, actualizare și ștergere (CRUD).
- Limitarea consumului maxim de energie pe oră pentru fiecare dispozitiv.
- Asocierea dispozitivelor cu utilizatorii corespunzători.
- Este responsabil pentru trimiterea deviceId-ului și a valorii maxime prin cozi spre Consumer.

Microserviciu pentru Consumer

- Este responsabil cu calcularea sumei și trimiterea unei notificări în cazul în care se depășește valoarea maximă.
- Consumul maxim pe oră va salva în baza de date.
- Obținerea informațiilor despre Device se obține prin două cozi care comunică cu Device Microservice.

Producer

- Este responsabil pentru citirea datelor din fișierul sensor.csv și trimiterea acestora printr-o coadă la consumer.
- Pornirea a două thread-uri cu ajutorul cărora se vor trimite datele.

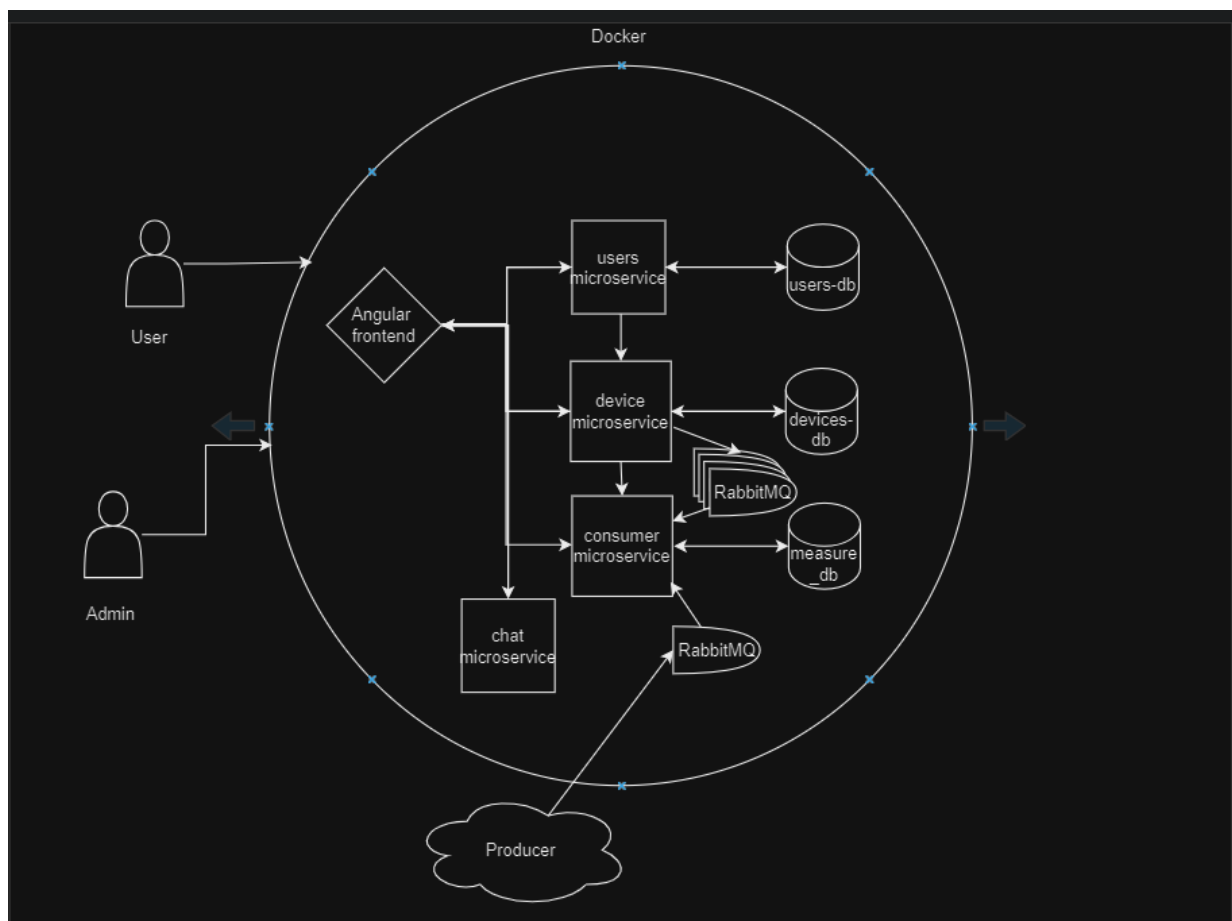
Chat

- Este responsabil pentru comunicarea între Admin și User. De fiecare dată când dorim să trimitem un mesaj acesta este trimis prin Web Socket, este interceptat de backend și este trimis spre destinație. Cu ajutorul unui endpoint primim mesajele de tipul typing, iar pentru mesajele de tipul read avem un endpoint separat care primește mesajele și le trimite spre destinație.

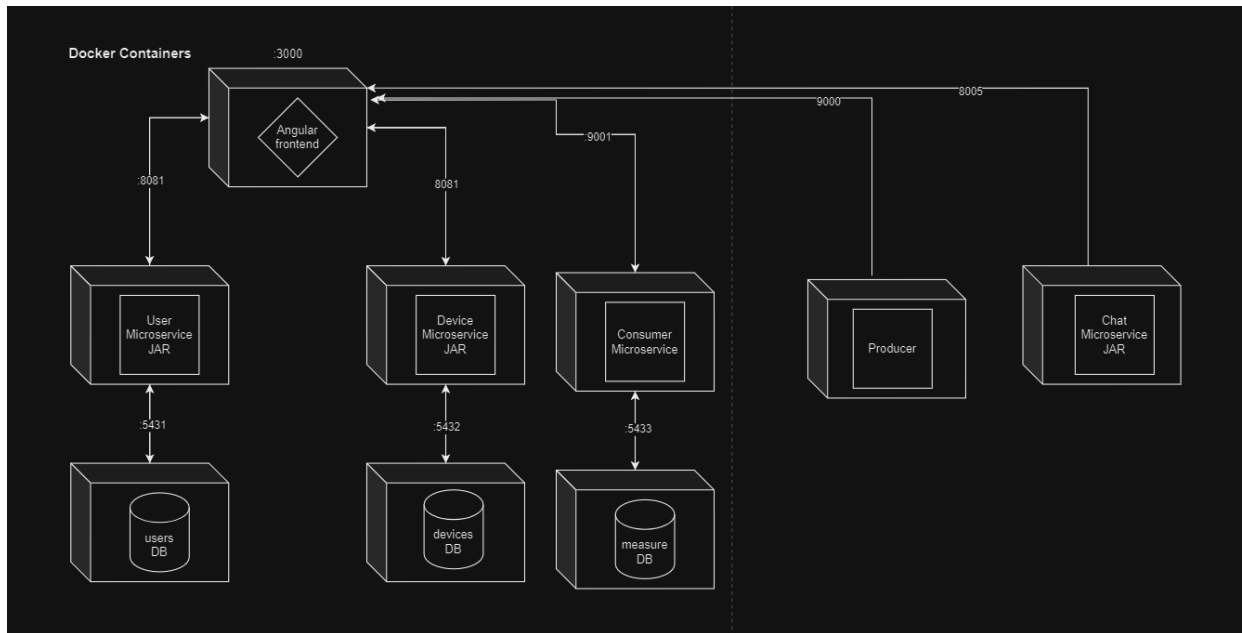
2. Implementare

Am creat Chat Microservice care se ocupa de conexiunea si comunicarea prin WebSocket. Pe frontend am creat doua chat-uri, unul pentru admin si altul pentru user. Adminul are posibilitatea sa selecteze dintr-o lista un utilizator, se va conecta pe canalul sau de socket si v-a putea trimite si primi mesaje spre si de la acel utilizator. Cand se trimite un mesaj acesta este trimis spre backend care cu ajutorul unui endpoint primeste mesajele si le trimite spre destinatar. De asemenea pentru typing cand se detecteaza cu metoda `OnInput()` se trimit mesaje de tipul typing celui alt utilizator, cand se primesc mesaje de tipul typing se pune pe `true` o variabila, iar in aceasta este pe `true` se afiseaza un mesaj „User is typing” sau „Admin is typing” in functie de caz. Pentru a vedea daca celalalt utilizator a vazut mesajul trimis am ales urmatoarea abordare: cand un utilizator primeste un mesaj trimite celui de la care a primit mesajul un mesaj de tipul `seen`, iar cand se primeste un mesaj de tipul `seen` variabila `isRead` va fi puse pe `true` si se va afisa mesajul „Seen” cand se va da click pe mesaj.

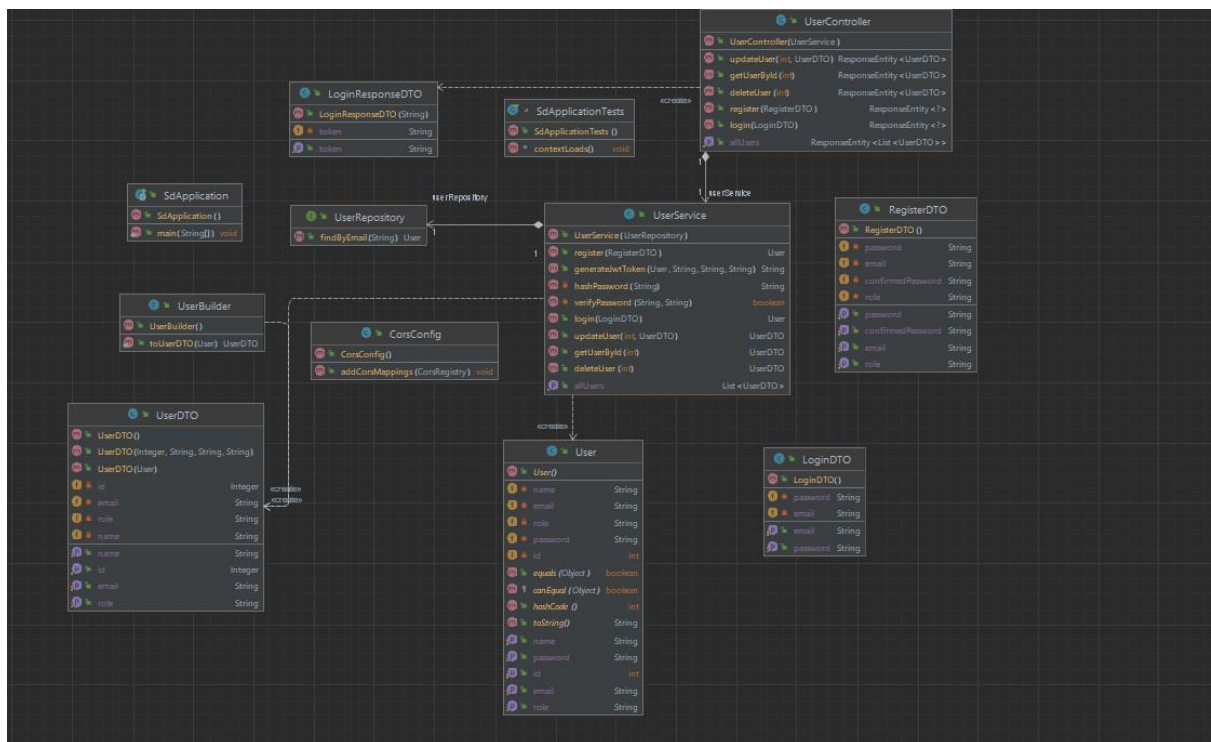
Pentru securitate am ales sa folosesc abordarea cu `jwt`. Astfel cand un user se inregistreaza ii salvam datele in baza de date, iar cand se logheaza se vor verifica datele, iar in cazul in care sunt valide se va genera un token care va fi salvat in local storage. Acest token va fi trimis in header la fiecare request. In functie de rol in navbar ii este prezentat ce are voie sa acceseze fiecare utilizator. Aceste optiuni sunt alese cu ajutorul clasele `authService` si `authGuard` care verifica si pun conditia pe fiecare link de catre cine pot fi accesate. Mereu se verifica daca `jwt` este valid si daca utilizatorul are rolul necesar, altfel v-a fi redirectionat spre pagina de login.



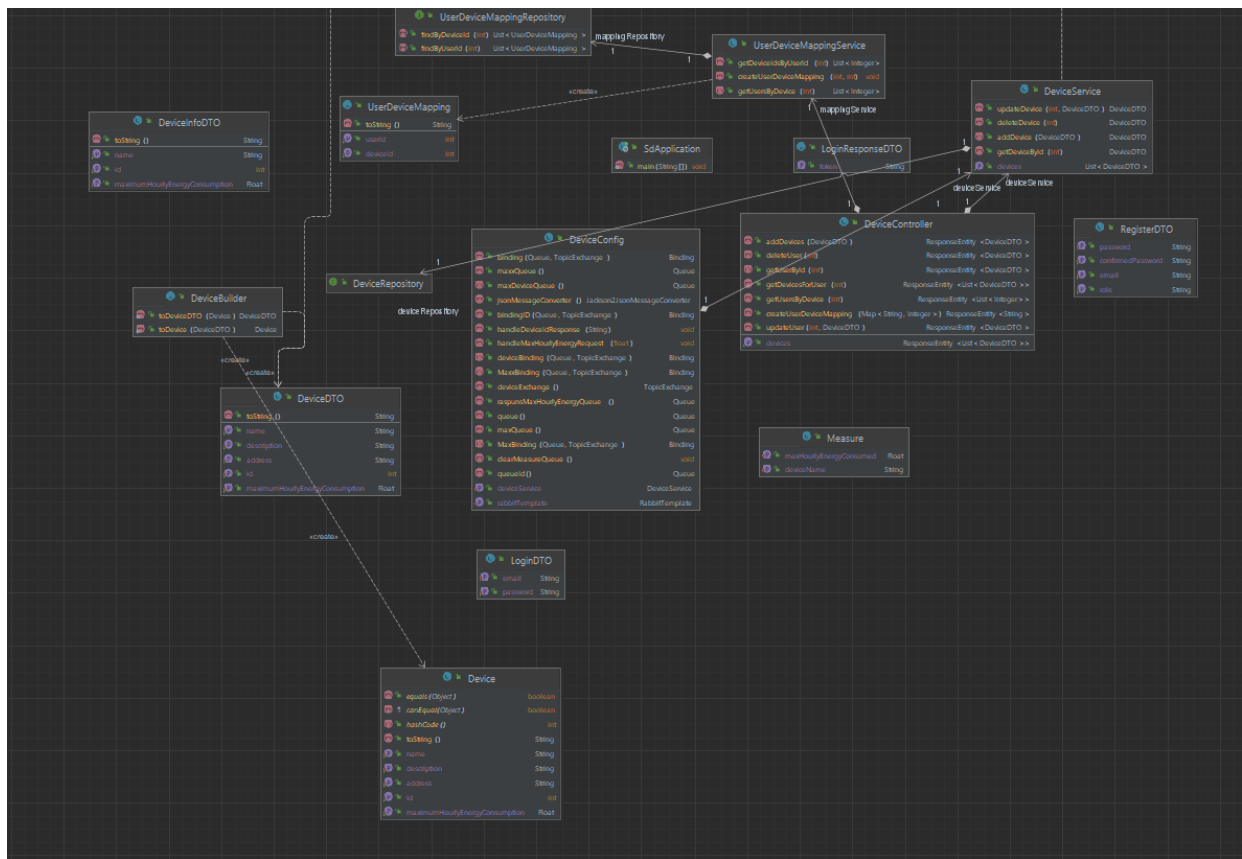
Figură 1. Conceptual Diagram



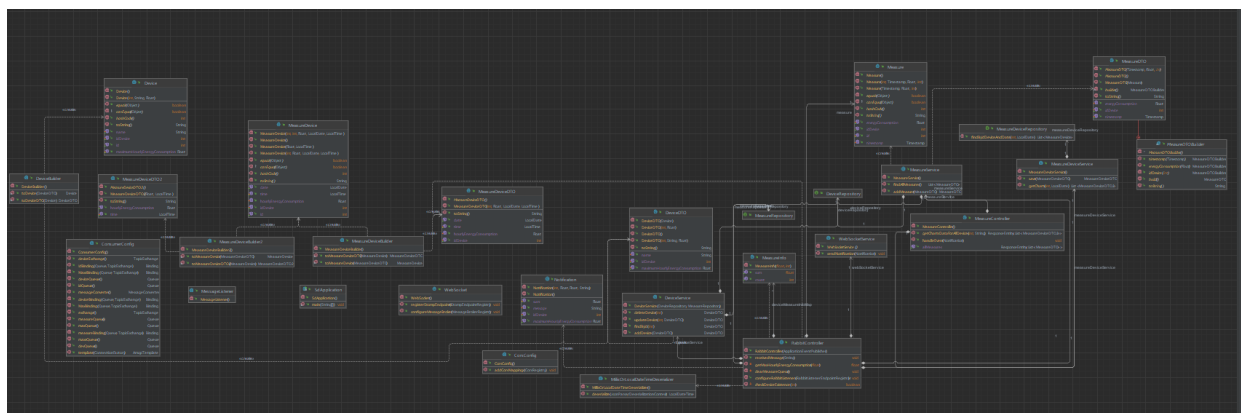
Figură 2.Docker Containers



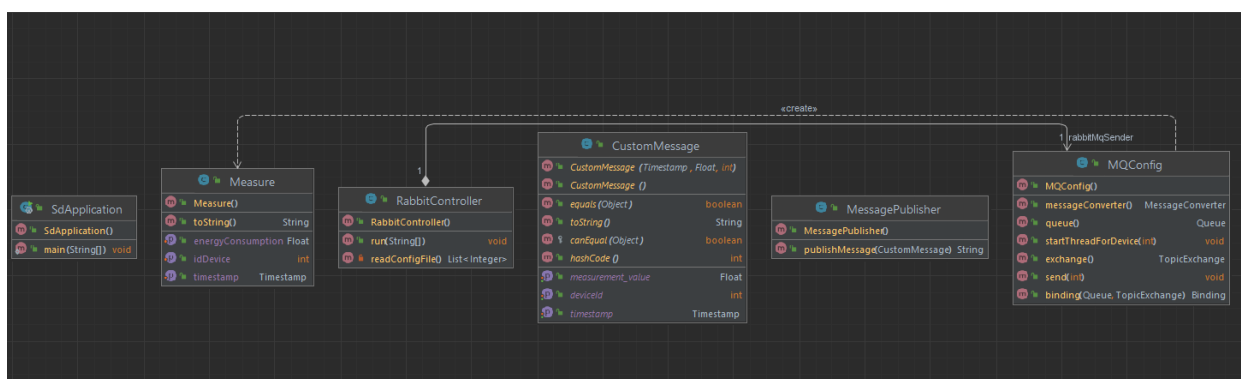
Figură 3.Diagrama UML User



Figură 4.Diagrama UML Device

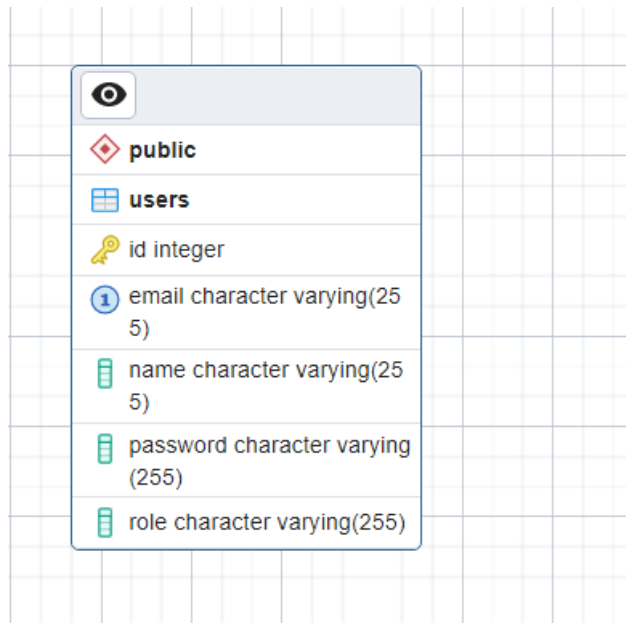


Figură 5.Diagrama UML Consumer

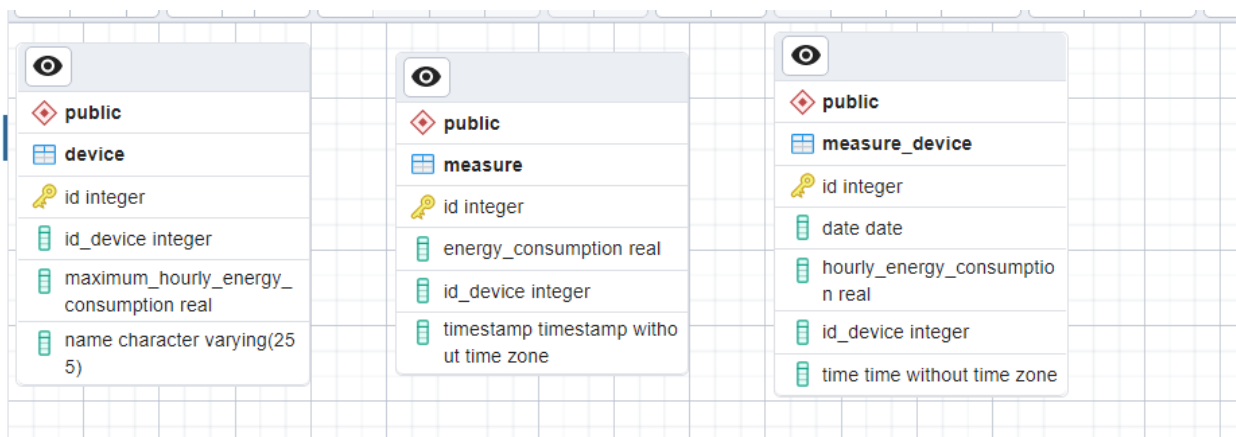


Figură 6.Diagrama UML Producer

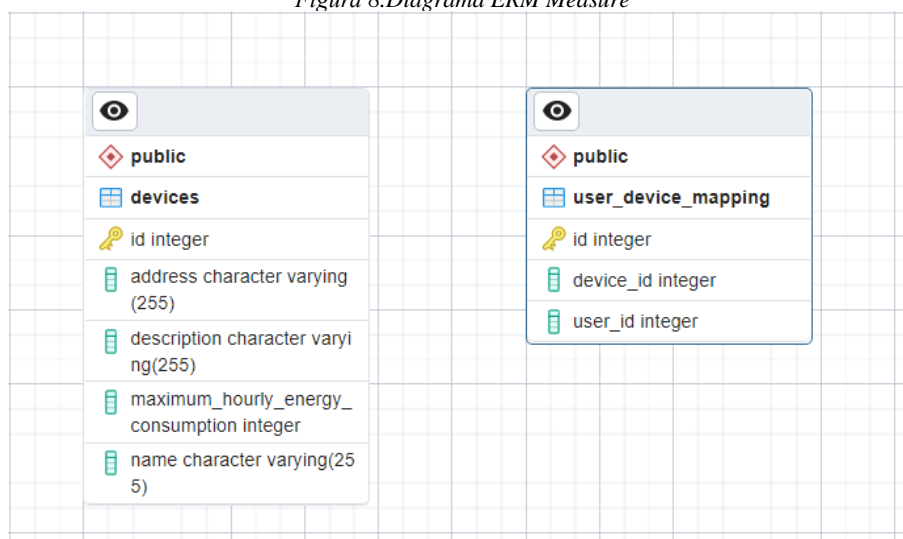




Figură 7. Diagrama ERM User



Figură 8. Diagrama ERM Measure



Figură 9. Diagrama ERM Device

3.Read Me

Comenzi

- `docker compose-up`
- `docker build -t frontend:latest .`
- `docker run --name frontend -ti -p 4200:80 frontend:latest`
- `docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.12-management`
`docker exec -it rabbitmq /bin/bash`

Considerații pentru Construire și Execuție

Frontend

Instalare și Pregătire:

Asigurați-vă că aveți Node.js și npm (Node Package Manager) instalate.

Instalați Angular CLI folosind următoarea comandă:

- `npm install -g @angular/cli`

Configurare și Pregătire pentru Proiect:

Navigați la directorul frontend și executați următoarele comenzi pentru a crea un proiect Angular și a instala dependențele:

- `ng new energy-management-app`
- `cd energy-management-app`
- `ng generate component component_name`

Pentru a rula aplicația aveți nevoie de comandă:

- `ng serve`

Device, User Microservices, Producer si Consumer, Chat

Asigurați-vă că aveți Java Spring REST si un IDE precum IntelliJ, proiectul de spring se poate genera pe site-ul Spring initializr.

Pentru a putea comunica intre microservicii utilizand RabbitMQ, trebuie sa deschideti Docker Desktop, iar intr-un PowerShell sa rulati comanda prezentata mai sus.

3. Bibliografie

- <https://start.spring.io/>
- <https://www.docker.com/>
- <https://spring.io/microservices>
- <https://www.npmjs.com/package/ngx-toastr>
- <https://www.rabbitmq.com/>
- <https://canvasjs.com/angular-charts/>
- <https://www.freecodecamp.org/news/how-to-make-bar-and-line-charts-using-chartjs-in-angular/>