Documentatie assignment 1

Sisteme distribuite

Ionaș Alex

1. Cerință

Se cere să se dezvolte un Sistem de Management al Energiei care să includă un frontend și două microservicii, cu scopul de a gestiona utilizatorii și dispozitivele lor de măsurare a energiei inteligente. Sistemul va fi accesibil de către două tipuri de utilizatori după un proces de autentificare: ca administrator (manager) și client. Se vor putea efectua operațiuni CRUD (Creare-Citire-Actualizare-Ștergere) asupra conturilor de utilizator (definite prin ID, nume, rol: admin/client), dispozitivelor de măsurare a energiei inteligente (definite prin ID, descriere, adresă, consum maxim de energie pe oră) și asupra asignării utilizatorilor la dispozitive (fiecare utilizator va putea deține unul sau mai multe dispozitive inteligente în locații diferite).

1. Diagrame de implementare

2.1 Arhitectura

O imagine care conține text, captură de ecran, diagramă, Font

Descriere generată automat

Sistemul constă în două API-uri REST cu frontend comun în Angular și backend în Java Spring. Angular gestionează interacțiunea cu utilizatorii, în timp ce Java Spring asigură gestionarea datelor și logicii de afaceri în spatele scenei. Această arhitectură modulară permite comunicarea eficientă între frontend și backend, permițând manipularea datelor în mod dinamic.

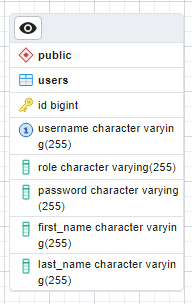
O imagine care conține text, diagramă, captură de ecran, Plan

Descriere generată automat2.2 Securitatea

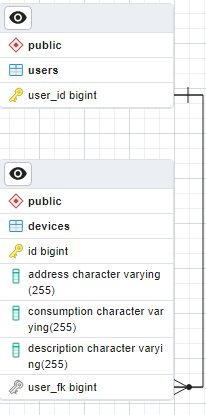
Securitatea a fost implementată prin intermediul Spring Security, care furnizează un filtru de securitate în lanțul de procesare al solicitărilor HTTP. Acest filtru asigură autentificarea și autorizarea utilizatorilor, validând cererile și gestionând fluxul de date criptate între client și server. Prin utilizarea mecanismelor avansate de autentificare, cum ar fi token-ul JWT și integrarea de filtre de securitate personalizabile, Spring Security oferă o protecție solidă împotriva atacurilor comune, cum ar fi CSRF, SQL injection și vulnerabilități de tip cross-site scripting (XSS). Acest sistem încorporează, de asemenea, politici de securitate complexe și un sistem robust de gestionare a sesiunilor, asigurând astfel un nivel ridicat de securitate și integritate a datelor în cadrul aplicației mele.

* 1. Bazele de date

1. pentru microserviciul care gestionează utilizatorii



1. pentru microserviciul care gestionează dispozitivele



1. Sincronizarea între cele două microservicii

Pentru a asigura o sincronizare eficientă între două microservicii, se utilizează apeluri API către cele două backend-uri, facilitând schimbul de informații între aceste componente. Astfel, se asigură consistența datelor și coerența acțiunilor între microservicii.

De exemplu, atunci când se creează un utilizator nou, se face un apel către backend-ul “users microservice”. În urma acestui apel, se raspunde cu o entitate de tip utilizator, care conține și id-ul acestei noi entități adăugate.

După ce s-a obținut id-ul proaspăt adăugat, se face un apel către backend-ul “devices microservice”, unde, în baza de date în tabela “users”, se adaugă doar id-ul noului utilizator. Astfel, la finalizarea operațiunii de adăugare a unui utilizator, ambele microservicii vor fi sincronizate. Pentru ștergere se procedează în mod similar.