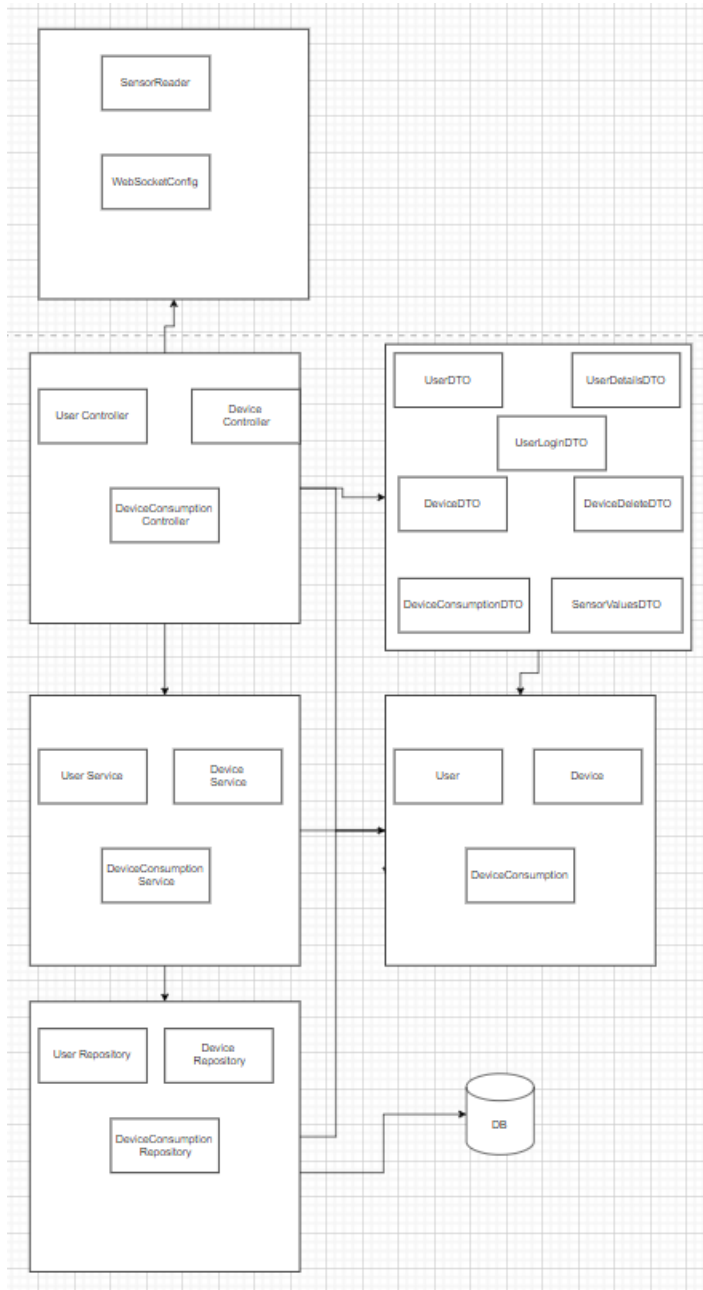


Documentatie Tema2

Tarta Manuel Vasile

Grupa 30641

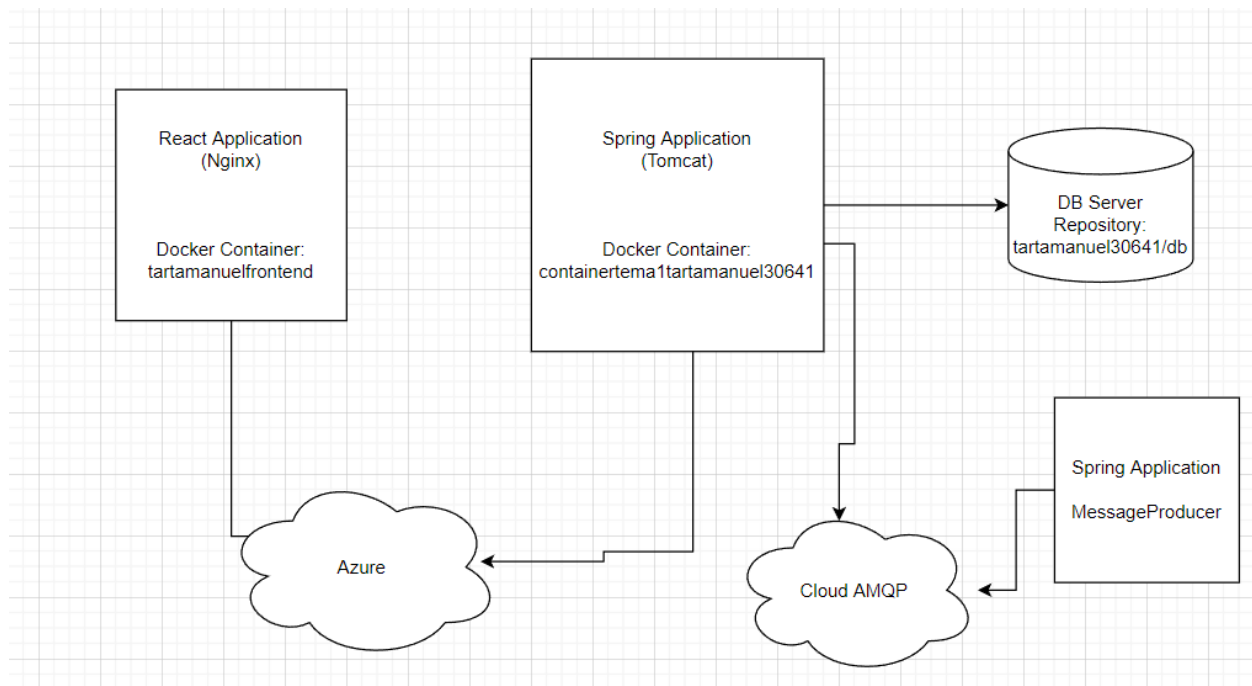
1. Arhitectura Conceptuala



In plus fata de tema 1 am mai adaugat un package care contine clasa `SensorReader` in care fac listen la coada din `CloudAMQP`. De fiecare data cand exista un element in coada, cu aceasta metoda il consum si il pun intr-o lista in controllerul `DeviceConsumptionController`. Acolo am o metoda `scheduled` sa ruleze din minut in minut (in loc de din ora in ora) in care grupez valorile care s-au pus in coada in acelasi minut, fac suma consumului si inserez in baza de date. Pot avea in coada oricate minute din acea ora, metoda va grupa elementele din secundele aceluiasi minut, va insera in baza de date o noua linie in tabela `DeviceConsumption`, va verifica daca suma a depasit maximul. Daca

suma este mai mare decat valoarea maxima pentru acel device, am si un websocket, care este configurat in clasa WebSocketConfig si care trimite un DTO (acelasi DTO pe care il folosesc si in coada rabbitMQ) pe un anumit canal de pe care va fi preluat in frontend. DTO-ul folosit contine valoarea citita din csv care reprezinta consumul device-ului in cele 10 minute(10 secunde in cazul meu), device-ul pentru care s-a facut citirea si timestamp-ul, adica data si ora la care s-a facut citirea. Trimit acest DTO spre frontend doar daca id-ul device-ului pentru care fac aceste inserari reprezinta un device al clientului care este logat in acest moment. Daca clientul logat nu are acest device, fac doar inserarea, nu trimit nicio notificare spre frontend. In frontend fac listen la canalul deschis prin intermediul websocketului, si oricand primesc un mesaj afisez un mesaj care contine, valoarea cu care s-a depasit maximul, id-ul device-ului si ora la care s-a produs depasirea de consum. Mesajul ramane pe pagina pana clientul se delogheaza, sau pana de refresh sau pana cand vine inca un mesaj care il suprascrie pe cel vechi.

2. Deployment Diagram



Fata de tema1, am adaugat o noua aplicatie, un message producer. Aceasta noua aplicatie are rolul de a citi date din csv o data la 10 secunde(in loc de 10 minute). Dupa ce le citeste, obtine si id-ul device-ului pentru care citeste aceste date si cu un dto trimite in coada un obiect ce contine valoarea citita, id-ul si timestamp-ul la care o facut citirea. Chiar daca aplicatia de backend nu este deschisa in acel moment, datele trimise vor ramane in coada pana vor putea fi consumate. Logica pe care am facut-o nu permite sa se poata face citiri deodata pentru mai multe dispozitive, deci la un moment dat in coada pot avea date doar pentru un singur dispozitiv. Ca sa se trimita obiectele in coada prima data sunt transformate in json, si la fel cand sunt primite sunt transformate din json inapoi in DTO.