**Frontend**

Use-case-k:

1. Elem felvitele naplóba:
   1. A felhasználó beír egy keresési kifejezést, majd a keresés gombra kattintva egy lista tölt be az arra illeszkedő elemek neveivel
   2. A felhasználó a felsorolásból kiválaszt egyet, ekkor megkapja annak tápanyag adatait
   3. A mennyiség megadása után az mentés gomb hatására bekerül a naplóba

**Mikroszolgáltatások architektúrája**

Azonosítás

API Gateway

Naplózás

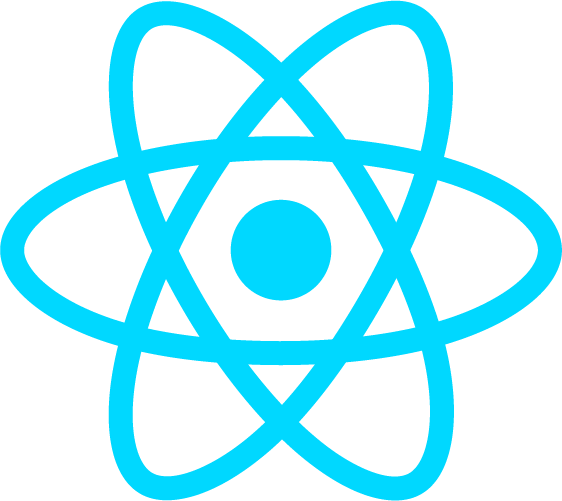
E-mail

Tápanyag adatok



MongoDB

MsSQL



Üzenetsor

MongoDB

**Kommunikáció**

* SPA ⬄ Naplózás
  + SPA 🡺 Naplózás
    - Naplóadatok lekérdezése
    - Új bejegyzés készítése
    - Bejegyzés szerkesztése
    - Bejegyzés törlése
  + Naplózás 🡺 SPA
    - Naplóadatok visszaadása
    - http válaszüzenetek
* SPA ⬄ Tápanyag adatok
  + SPA 🡺 Tápanyag adatok
    - Keresési szó search végpontra
    - Elem név data végpontra
  + Tápanyag adatok 🡺 SPA
    - JSON név-id párokkal, amire illeszkedik a keresés
    - JSON név-id pár tápanyag lista tartalommal
* SPA ⬄ Azonosítás
  + SPA 🡺 Azonosítás
  + Azonosítás 🡺 SPA
* SPA ⬄ E-mail
  + SPA 🡺 E-mail
  + E-mail 🡺 SPA

**Nutrient\_info\_service**

Technológiák:

* ASP.NET Core
* MongoDB

Use-case-k:

1. Név szerinti keresés: a megadott szövegre illeszkedő (a külső api által „hints” mezőben visszaadott) elemek felsorolása, neveinek visszaadása
2. Kiválasztott név szerinti tápanyag lekérdezés (pontosan egyik elem a felsoroltak közül): elem adatainak lementése a lokális adatbázisba, a tápanyagértékek visszaadása

Architektúra:

Rest interfész

Üzleti réteg

Adatelérési réteg

Külső adatbázis

Belső adatbázis

* Rest interfész
  + Rest stílusú kéréseket fogad és szolgál ki
  + Végpontjain fogad: név szerinti keresést és név szerinti kiválasztást
  + Végpontjain visszaad: keresési eredmény listát, kiválasztás eredmény tápanyag adatokat
* Üzleti réteg
  + Kérések logikai kiszolgálását végzi, kapcsolat az interfész és az adatelérési réteg között
  + Feladatai: ha keresés érkezik, akkor a külső adatbázist elérő függvénnyel kéri le a névsort és adja eredményül; ha kiválasztást történt, akkor a belső adatbázist elérő függvénnyel próbálja meg a lekérést, de amennyiben ez üres, akkor a külső adatbázishoz fordul
* Adatelérési réteg
  + Az adatbázisok és az üzleti réteg között teremt kapcsolatot, ismeri az adatbázisok szerkezetét, technológiáját, ezek eléréséhez nyújt független elérést kifelé
  + Feladatai: külső adatbázis lekérdezése API-n keresztül, belső adatbázis lekérdezése, belső adatbázis szerkesztése
* Belső adatbázisű
  + MongoDB alapú adatbázis, melyben az egyes elemekhez tartozó tápanyag adatok vannak lementve
  + Lényegében cache funkciót lát el
* Külső adatbázis
  + API-n keresztül elérhető adatbank, amelyből a tápanyag adatok érkeznek
  + <https://developer.edamam.com/food-database-api>

**Adatbázisok felépítése**

* Felhasználói naplók
  + MongoDB
* Tápanyag adatok
  + MongoDB
  + Példa

{

"name": „Apple”,

"id": „food\_a1gb9ubb72c7snbuxr3weagwv0dd”,

"kcal": 52.0,

"protein": 0.26,

"fat": 0.17,

"carbs": 13.81,

"fiber": 2.4

}

* Azonosítás (felhasználói adatok)
  + MsSQL

**Referenciák**

<https://medium.com/aspnetrun/build-catalog-microservice-using-asp-net-core-mongodb-and-docker-container-88b8fd4d5040>