

Felhőszámítási rendszerek - féléves beadandó dokumentáció  
Óbudai egyetem - NIK - NIXCC1HMNE/CC1<sub>E</sub>A<sub>M</sub>S<sub>N</sub>

Burian Sándor

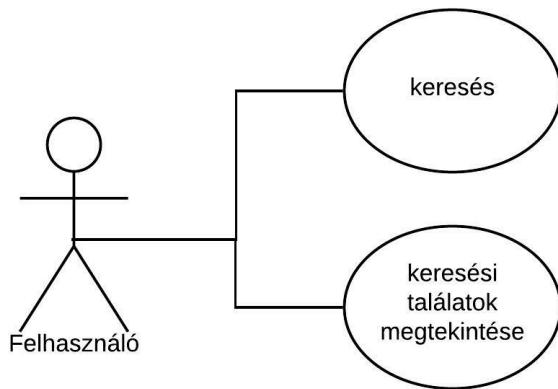
2021, Október 24

## *Téma bejelentő*

A féléves beadandóm a BSc-s szakdolgozatom során elkészült oktató oldal[4] továbbfejlesztése, kiegészítése egy egészségügyi adatbázis keresővel, amiben különböző APIkat integrálok, illetve az egyes apival nem rendelkező adatbázisokhoz egy crawler segítségével férek hozzá. Mindezt amennyiben megoldható a Jina AI[1]-val egészíteném ki, de még ennek jogi feltételei nem egyértelműek. Megvalósítani Azure vagy AWS rendszeren tervezem, mivel server oldali scriptekhez szükségem van futtatási jogosultságra, a program nyelve ennek függvényében változhat.

# *Bevezető*

A féléves beadandóm a BSc-s szakdolgozatom során elkészült oktató oldal[4] továbbfejlesztése, kiegészítése egy egészségügyi adatbázis keresővel, amiben különböző egészségügyi témaikra kereshetünk rá és OLAP szerű hasonló eredményt kapunk. A dolgozat csak magát a kiegészítést tartalmazza, mivel jelenleg csak az az érdekes számunkra. Munkámban több különböző technológiai megoldást fogok megmutatni több különböző technológiára amivel adatokat nyerhetünk ki adatbázisokból. Az egyik egy közvetlen API, a másik egy lokálisan frissített index tábla, a harmadik egy crowllerrel webscrappelt adathalmaz. A feladatot AWS EC2 használatával oldottam meg, melynek konfigurálására kitérek az következőkben. Próbálkoztam más adatbázisok integrálással is, azokról is szót ejtek, valamint továbbfejlesztési lehetőségeket is felvázolok.



1. ábra. A program use case diagramma

# Felhasználói dokumentáció



2. ábra. A főoldalon beadhatjuk a kreszt kifejezést

3. ábra. A találatokat megjelenítő felület



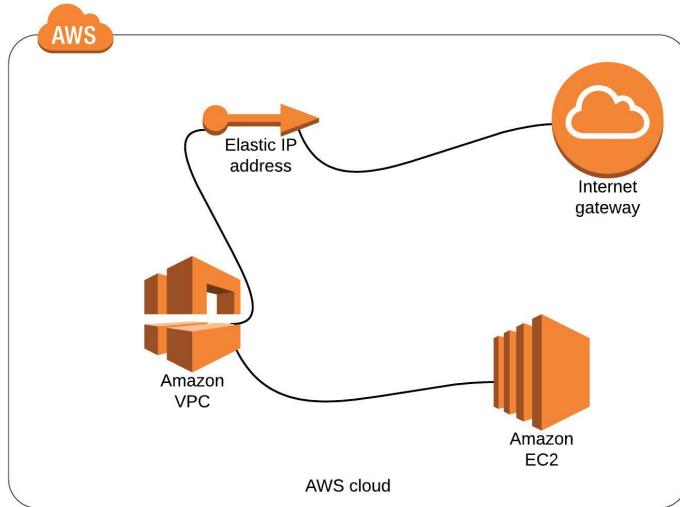
4. ábra. A találatokat ki is kapcsolhatjuk, keresési forrássokként, illetve a logomellett megjelenő ikon jelzi, hogy még töltődik (sárga), betöltődött(zöld), hiba történt(piros) a lekérdezés az adott forrásból.

# *Fejlesztői dokumentáció*

## 0.1. Futtatási körülmények létrehozása

### 0.1.1. Környezet

A feladatot Amazon Elastic Compute Cloudon oldottam meg, ahol Amazon linux gépet telepítettem, majd arra telepítettem Apache webservert, PHP7-et, Python3-at, PIP-et, majdazzal pár hasznos Python segédeszközt. Ehhez első lépésként



5. ábra. A program felépítése az AWS szemszögéből

### 0.1.2. VPC

Ehhez először egy virtuális tűzfalat hoztam létre és egy biztonsági csomag beállítást, ahol a név és egyéb metadata után az elérés módjait állítottam be. SSH/HTTP protokollokat engedélyeztem [5].

### **0.1.3. EC2**

Ezután magát a virtuális gépet hoztam létre egy Amazon Linux alapú isoval. Megadtam a hardver engedélyeket, társítottam a korábban létrehozott VPCvel, és a networkingot automatikusan kigeneráltattam az Amazonnal. Ezt követően az ec2-userel már be is lehet SSHzni, telepíteni az Apacheot és egyebeket[3].

## **0.2. Használt technikák**

Mint az ezekeből is kitűnt, Innentől bármilyen módszert használhatunk szabadon. Én így is tettek, minden egyes adatbázishoz külön-külön írtam egy megoldást ami az adatokat összeszedi és lekérdezhetővé teszi.

Segédprogramként telepítettem az Apcheon kívül PHPt, Python, PIP-et.

Hogy a kapott keresési eredmények OLAP-hoz méltóan táblázatként letölthetőek legyenek integráltam egy excel táblává átalakító javascriptet [<https://codepedia.info/javascript-export-html-table-data-to-excel>].

A program felhasználói felületéhez Zurb Foundation CSS rendszerét használtam.

### **0.2.1. API alapú keresés példa**

Az API alapú datlekérdezés az egyik legegyszerűbb feladat így ezzel kezdtem. Állatorvosi lónak egy korábbi projektem választottam, a Magyar Koronavírus Info oldal APIját.

Ez az oldal egy hírgyűjtő oldal ami RSS alapon különböző magyar nyelvű híreket gyűjt a koronavírussal, vagy azzal bélyegzett hírekkel kapcsolatban. Az APIIn keresztül frissíthető is az adatbázis valamint lekérdezhető az elmúlt 24 óra vagy az összes hír is. Én természetesen utóbbt használom, hisz a kereső felhasználó is erre számít. A felhasználó számára megjelenik az összes hírből az amiben substring alapú egyezés van, majd az ehhez tartozó hír címe, leírása, és link a hírhez, minden egy táblázatban megjelenítve, ami javascript segítségével letölthető Excel fájként is.

### **0.2.2. Indexfájl alapú keresés**

Erre a feladatra az Academic Torrents adatbázist választottam. Nekik ugyans szerepel egy XML alapú indexfajljuk amit időről-időre frissítnek amiben az összes elérhető torrentfájl linkje, leírása, címe elérhető. Kérésükre nem az oldalukon kérdezem le, hanem letöltve a fájlt lokálisan keresek benne, a az indexfájlt pedig havonta frissítem.

### **0.2.3. Webscrap alapú keresés - data.europa.eu**

Ez egy érdekes feladat amihhez a data.europa.eu-t szerettem volna használni, de mint kiderült az oldal bugos, ezért sok esetben kapnék false értékeket. Innen érdemes a kereséshez generált RSS/ATOM fájlokat célba venni, mivel a GUI alapú válasz lusta betöltést vagy AJAXot használ ezért a használt népzerű python segédfüggvénygyűjtemény, a Beautiful Soup szempontjából nem igazán kezelhető.

#### **0.2.4. Webscrap alapú keresés - eurostat (ec.europa.eu)**

Sajnos az adatbázisukhoz egy-egy apit kell kézzel legénárlni külön-külön minden adatsohoz így ezt nem tettek meg, inkább a tartalomból szedtem ki egy script ssegítségével a leíráshoz tartozó linkek[2]. Ezzel a módszerrel a teljes eurostat adatbázisa letölthető.

Az adatbázisban való keresést egy segédscript oldothatná meg mivel a Jina.AI nem teljesen alkalmas erre ezért úgy kell étrehoznunk a sajátkereső scriptünket, hogy az az eurostat összes belső kódját értlemezze, pl *HU* megfelelhető legyen *Magyarországnak* és így tovább.

#### **0.2.5. További fejlesztési lehetőségek**

Természetesen az egyértelmű továbbfejlesztési lehetőségeken kívül, hogy többadatbázst kötünk be (ezekre javaslat az oldal láblécében) képesek vagyunk mondjuk fordítót implementálni, és keresni vele több nyelven egyszerre, így ha a felhasználó beírja *Magyarország* akkor egyszerre kapjon találatokat a *Hungary* kifejezésre is. Erre esetleges lehetőség a Microsoft Azure rendszerében található fordító.

## *Irodalom*

- [1] Jina AI. *Jina AI - a Neural Search Company*. en. URL: <https://jina.ai/> (elérés dátuma 2021. 10. 18.).
- [2] gabboraron. *Felhőszámítási rendszerek*. original-date: 2021-09-07T09:53:13Z. 2021. nov. URL: [https://github.com/gabboraron/felhoszamitasi\\_rendszerek](https://github.com/gabboraron/felhoszamitasi_rendszerek) (elérés dátuma 2021. 11. 28.).
- [3] *General prerequisites for connecting to your instance - Amazon Elastic Compute Cloud*. URL: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connection-prereqs.html#connection-prereqs-get-info-about-instance> (elérés dátuma 2021. 11. 04.).
- [4] Burian Sándor. “”Tananyagok megosztását és tanulást segítő oktatóoldal””. (2020. jún.). Accepted: 2021-05-19T10:02:00Z. URL: <https://edit.elte.hu/xmlui/handle/10831/56291> (elérés dátuma 2021. 10. 24.).
- [5] *Tutorial: Create an Amazon VPC for use with a DB instance - Amazon Relational Database Service*. URL: [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateVPC.SecurityGroupEC2](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.SecurityGroupEC2) (elérés dátuma 2021. 11. 05.).