



Két heti beszámoló + dokumentáció

i.sz. 2k19. Július 12.

Kovács Máté

ELTE Szakmai gyakorlat

Első két hét részletesebb felbontása:

Első hét (2019.07.01-05)

Az első héten a SAP Hana szerver megismerésére szántuk csoportosan. Sajnos az SAP gyorsan fejlődő rendszere miatt a kapott tutorialok már elavultak voltak, de gyors korrigáció után már a megfelelő tutorialokat tudtuk böngészni mindannyian. A tutorialok magába foglalták az SAP fájlrendszerének használatát, azonkívül betekintést nyertünk a különböző projectek összekötésének how-to-jába.

Az SAP keretrendszer megismerés a hét végére átcsapott fakultatívvá, tekintve mindenki elkezdett "szakosodni" az általa választott feladatkörben. Én a Java programozási nyelv megismerését tűztem ki célul, több primitív konzol- és GUI alkalmazást is készítettem, melyekkel sikerült elsajátítani a nyelv alapvető szintaktikáját.

A hét végén megkaptuk az első feladatunkat, mely lentebb van részletesen leírva. A feladat megoldása eleinte minden személynél eltért, mivel az adatbázis tisztításához mindenki máshogy állt hozzá.

Külön Problémát okozott az SAP szerverek túlterheltsége/leállása, emiatt a pénteki nap nagyrészt eredménytelenül telt el.

Második hét(2019.07.08-12)

Hétfő

A második héten már tudtuk használni az ELTE által futatott házi SAP hana webszerveret, Hétfő délután már teljes jogosultságunk volt hozzá, ezzel újból nekiállhattunk tisztítani a kapott adatbázist(lentebb részletezve).

Kedd (Szerda, csütörtök, péntek nyaraláson vettem részt, hétvégén hoztam be a lemaradást)

Reggel Bélecski András segítségével a teljesen elérhető odata-ban dolgoztunk. Déltől kezdve felbátorodva azon az ötleten, miszerint a kapott adatokból a cégeket el lehetne helyezni egy beágyazott map api-n, nekiálltam a feladat megoldásának Java környezetben. A feladatot tovább nehezítette a tény, hogy :

- Google maps API nem ingyenes, ezáltal későbbi javaslat után Bing maps API-t kellett használnunk
- a Bing maps API jóval fejletlenebb és kevésbé felhasználóbarát Java környezetben

-Java környezetben ilyenfajta problémát előttem még nagyon kevés ember oldott meg, melyet általában nem dokumentált az interneten, ezáltal "ismeretlen vizeken hajóztam".

Hosszú keresés után végre találtam egy oldalt, ahol a probléma megoldása hasonló volt ahhoz amire nekem szükségem van.(Sajnos az oldalt a mai napig nem sikerült újra megtalálni.)

A neten talált kód alapján sikerült létrehozni egy olyan programot, mely a bing api-t használva, megfelelő mennyiségű helyadat(házszám, utca, város, ország) birtokában egy bing szerveren tárolt xml/json file-t ad vissza.Az itt felmerülő probléma az volt, hogy a visszakapott xml minden karaktere a két koordinátán kívül felesleges, 'trash' volt, melytől a megszabadulás újabb problémákat vetett fel.Eleinte próbálkoztunk az xml/json file parse-olásával, ezáltal könnyen kinyerhetővé és hivatkozhatóvá váltak volna a koordináták. Következő probléma egy, előre megírt parser beimportálása volt a projektbe, mely jelentős ideig (egy-másfél nap) eltartott, mivel az online talált parserek mind 'homebrew', third party által készített parserek voltak, amik használata és megértése kisebb-nagyobb nehézségeket vetett fel(végre megértettem ezáltal a helyes és kiterjedt dokumentáció lényegét). Ellenben hamar rájöttem, hogy a bing szerver oldali xml file szintaxisa eltér a parser által várt szintaxistól, így kénytelen voltam saját parsert írni, mely működése meglepően egyszerű: Egy egyszerű keresés tételt (<http://bzsr.web.elte.hu/progmod2/konyv.pdf>, 143.o. 12.2.5 ös bekezdés) módosítottam úgy, hogy a .json file-on vesszőnként splitelve, stringeket ellenőrizve haladjon végig, és így találja meg a megfelelő adattagot.Mivel a keresés nem megy végig a fileon, emiatt a saját "parser"-em gyorsabb, mint bárki más által írt parser, feladatspecifikussága miatt.Sajnos a program működése még így is elfogadhatatlanul lassú, főleg a webes lekérdezés lassúsága miatt, mely jelentősen függ a hálózat sebességétől, a szerveroldal terheltségétől, stb., és ezáltal más programba beágyazva, 1000+ lekérdezésnél már észrevehető várakozási idő.

Ezt a problémát figyelembe véve döntöttem a Java nyelv időleges félretételén, és kezdtem el az Angular környezet és JS nyelv felületes tanulmányozását, mellyel töltöttem el a maradék időmet a hétvégéből.

Feladat megoldásának áttekintése

2019, es ELTE-Soft nyári szakmai gyakorlatának első két hete.

Az első két hét feladata ismeretterjesztő jellegű volt, hiszen mindannyian teljesen új területre sodródtunk, egy adatbázis rendszert kellett megvalósítanunk az SAP-HANA adatbáziskezelőben, majd az adatbázis odata formátumban elérve, egy front-end oldali diagram megvalósítása volt a cél.

Célok

1. egy adatbázis rendszert létrehozni SAP-HANA adatbáziskezelőben
2. odata formátumban elérve, egy front-end oldali diagram megvalósítása

Tervezet

1. Adatbázis
 - a. Adott csv fájl feltöltése a SAP Web IDE-be , majd adattisztítás és normalizálás, amennyire csak lehetséges
 - b. Ezután kalkulációs nézet létrehozása
 - c. Majd Odata-ban exportálni.
2. Front-end
 - a. Odata elérése
 - b. Adatok vizualizációja

Megvalósítás menete

Adatbázis

Egy eléggé rossz csv fájlt kaptunk kiindulópontnak, hiszen tele volt sorduplikátumokkal, összeférhetetlen adatokkal, valamint helytelenül kitöltött sor is található volt benne. Ezek kiszűrése kisebb fejfájást okozott de előbb utóbb mindenkinek sikerült megoldást találni a problémára. Én személy szerint próbálkoztam azzal, hogy töröltem azokat a sorokat amelyekben volt üres adatmező, vagy hibás típusú adat volt valahol, emellett töröltem a duplikátumokat egy sima distinct lekérdezéssel.

A csv fájl feltöltése igen egyszerű volt, manuálisan annyit kellett megtenni, hogy a projekthez, hozzáadunk egy database module-t majd a .hdbcds kiterjesztésű fájlba kézzel beleírni az adattábla nevét és sorainak attribútumait: pl

```
entity regio_kod {
    regio_kod : Integer;
    regio      : String(600);
};
```

Ezután lehetett is a táblába importálni a csv-t.

Miután ez megvolt, kezdődhetett a tábla darabokra vágása, külön táblába szedtem mindent aminek volt, kód attribútuma (így tűnt logikusnak).

És jöhet a calc view: miután megvolt a tábla értelmesen elvégzett szétDarabolása eme tutorial segítségével a Kalkulációs nézet megteremtése pofonegyszerű volt (csak kicsit macerás a sok húzogatás) <https://developers.sap.com/tutorials/xsa-graphical-cube-view.html>

És az odata : a projekthez hozzá kellett adni egy új modult, : 'new nodejs module'

És engedélyezni kellett az xsjs supportot benne létrehozáskor. Majd az újonnan készült moduleban létrehozni egy .dxodata kiterjesztésű fájlt és abba a következő kódrészlet megoldotta az exportot:

```
service{  
  "Calc_view" as "DATA" keys generate local "ID" ;  
  "db.data" as "ALL" keys generate local "ID" ;  
  "TARSASAGOKORIGIN" as "TARS" keys generate local "ID";  
}
```

Front end

Angular oldal készítését vállaltam el, hiszen ajánlották, hogy jó lenne ha valaki foglalkozna angularral.

Legfontosabb kérdés, az volt, hogy hogyan érem el az eddig készített Odata exportom. Erre egyszerű megoldást találtunk : `http.get(odata url)` segítségével szuper egyszerűen megoldható. (ELTEkintettünk attól a problémától, hogy csak 1+e3 sort tudunk egyszerre lekérdezni, most ez is jó volt nekünk.)

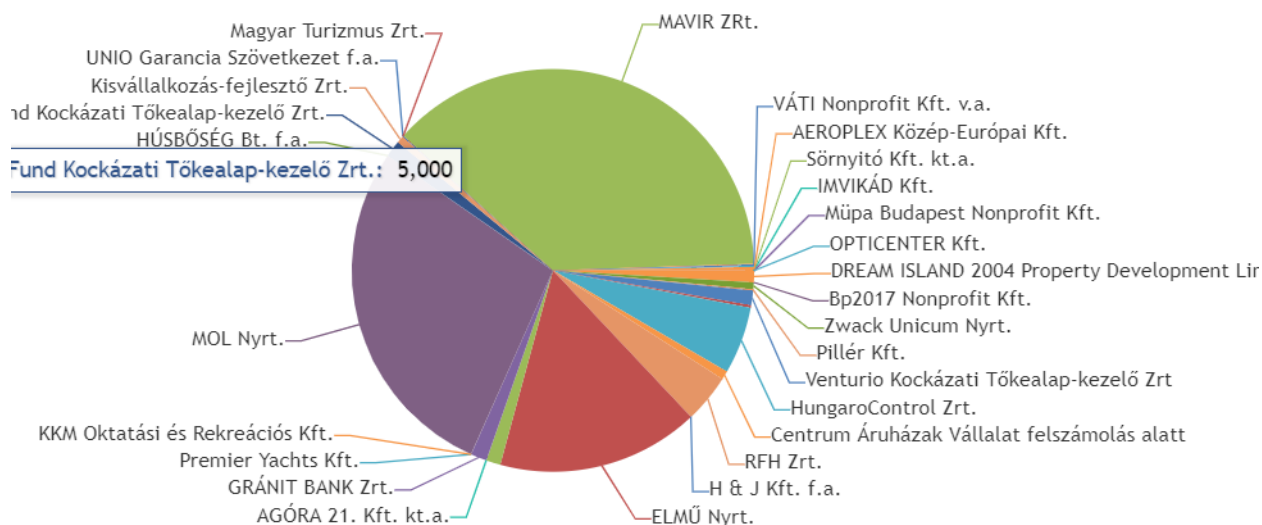
Ehhez ez a tutorial oldal tökéletes magyarázatot nyújtott:

<https://angular.io/guide/http>

Ez tökéletes is volt csak, létezik egy **same-origin policy** ami eléggé nagy fejfájást okozott, de proxyval ki lehetett kerülni.

Ezután már meg is voltak az adatok amiket js- oldalon könnyű formázni és kimutatni, kerestünk is egy lib et ami pont ezzel foglalkozik, így találtam rá a canvasjs-re mellyel gyönyörűbbnél gyönyörűbb ábrákat lehetett készíteni, így kész is lett a kitűzött feladat:

Cégek vagyona összvagyon:373608 (1e6 HUF) társaságdb: 52



2019.07.15-től végzett munka beszámolója

07.15(hétfő)

Eredetileg már pénteken bemutatásra kerültek volna a miniprojektek, ennek megfelelően pénteken már el is készült minden ránk kiszabott feladat. A mai napon a dokumentációt szépítettük, Emellett a projektben esztétikai változtatásokat végeztem el(html, css módosítása).

07.16(Kedd)

A nap az angular felszínebb megismerésével telt el.A feladatot megnehezítette a javascript nyelv ismeretének hiánya, de keresés után az angular oldalán talált tutorialok alapján sikerült lépéseket tenni a fejlesztői környezet felületes elsajátításához.

07.17(szerda)

A mai napon bemutattuk az első két és fél hétben készített projektet.Mivel két projektben is részt vettem, úgy döntöttem hogy inkább a Java-ban készített programomat ismertetem a bemutatón, míg az angular-os projektet csak

említettem, mivel annak a teljes dokumentációja megtalálható feljebb. A nap nagy része a bemutatóval telt el, itt megbeszélhettük ki milyen módon oldotta meg a kiszabott feladatot, ötleteket vehettünk át egymástól az elkövetkezendő további négy hétre.

07.18(csütörtök)

Nekiálltam a Java nyelv mélyebb megismeréséhez, Ebben az ELTE IK-s "programozási nyelvek java" kurzus segédanyagait és az interneten fellelhető tutorialokat használtam. Szerencsére a Java és az általam már korábban egész jól elsajátított C# nyelv hasonlósága megkönnyítette a feladatot. Külön nehézséget okoztak a következő hetek alatt a package-ek használata, melyek gyakran tartalom szempontjából lefedték egymást, így fölösleges tárhelyet foglalt a program, amely egy nagyobb projektnél már megengedhetetlen lenne.ű

07.19(péntek)

A nap ugyancsak a Java programozási nyelv megismerésével telt el, ezen belül is a dinamikus tömbök, konstruktorok, és ezek túltelhelése volt a fő téma, emellett a C++ban megszokott vektor helyett a Java által preferált List container típus megismerését tűztem ki célul, mely további vizsgálat után sokkal hasznosabbnak és könnyebben kezelhetőnek bizonyult mint az előbb említett vektor típus.

07.22(hétfő)

Ma ugyancsak a Java-val foglalkoztam, ezen belül a Java frontend moduljait vettem górcső alá. Míg valamivel egyszerűbben használhatóak a különböző triggerek a Java-ban mint a hasonlóan működő C# nyelv alatt, sajnos a Visual Studio eszköztárával és felhasználóbarát felületével sajnos nem tudja felvenni a versenyt.

07.23(Kedd)

A mai napon, konkrét projekt hiányában úgy döntöttem hogy eddig szerzett tudásomat felhasználva egy útkeresőt és mapgenerátort készítek, ezen miniprojekt megvalósításával telt a következő másfél hét.

07.24(Szerda)

Nekiálltam a projekt rekurzív módszerrel való megvalósításának.

Egy bináris karaktereket tartalmazó mátrixot használtam "pályaként", és ennek a közepére egy fixen letett útdarabtól kezdve kezdtem el felépíteni a gráfom. A rekurzív módszer, míg igen egyszerűen megírható volt, sajnos a több szálon való futtatás hiánya miatt időigényesnek bizonyult, emiatt a holnapi napra a feladat iteratív megoldását tűztem ki célul.

07.25(Csütörtök)

A projekt átírása iteratív módszerre időigényesnek bizonyult, mivel nem szándékoztam egy új projektet létrehozni neki, hanem az idáig írt programot akartam átalakítani. Sajnos ez által több időt elvesztettem, mintha egy új projektet nyitottam volna neki, de a nap végére volt egy működőképes iteratív megoldásom. A program két "duplaciklusban"(ciklus a ciklusban) ment végig a mátrixon, először középről felfele, majd középről indulva lefele. A ciklusban ha a mátrix indexének bármelyik szomszédja tartalmazott "utat" akkor ide is bizonyos százalék eséllyel utat rakott. A program egyértelmű hibáját, miszerint ez által általában csak a mátrix bal oldala (a ciklus balról jobbra halad) tartalmazott utakat, ekkor még nem vettem észre.

07.26(Péntek)

Mivel a programnak nem volt semmiféle megkötése arra hogy egymás mellé 2x2, 3x3, vagy 4x4 darab utat letehessen, ennek megakadályozását tűztem ki a mai nap céljának. Míg az írt kódsor jól működött elég hamar, ennek finomhangolása, különböző százalékos esélyek kipróbálása a nap további részét teljesen elvette.

07.29(hétfő)

A mai napon észrevettem a fentebb említett problémát, miszerint a gráf csak az mátrix egyik oldalát foglalja el. Ennek az orvoslására létrehoztam további két "duplaciklust", és a korábbi 2 helyett négy részre osztottam a mátrixot. Ezek után a készített gráfok ténylegesen véletlenszerű alakzatokat hoztak létre.

07.30(kedd)

Elkezdtem az útkereső algoritmus írását. Az elérhető algoritmusok közül úgy döntöttem hogy egy primitívebb A*-ot fogok elkészíteni, mivel az élek nem voltak

súlyozva (mátrix miatt).Az algoritmus elkészítése és a programon való "barkácsolás"-sal töltöttem el a napot.

07.31(szerda)

A mai napon beszámolót tartottunk Vincellér Zoltán tanár úrnak az eddigi munkánkról.A tanár úr adott egy projektötletet, mely szenzorok adatainak felhasználását/rendszerezését/feldolgozását jelentette.A feldobott projektötlet példák között volt egy ami különösen tetszett(Egy szenzorokkal felszerelt parkoló szimulálása, mely jelzi a szabad helyeket, és oda navigálja a személygépjárművet), a feladat megoldásának kutatásával töltöttem el további két napot, ellenben pénteken barátaim nyomására sajnos beláttam hogy egyedül nem tudom megcsinálni a kiszabott határidőig, és beszálltam az ő projektükbe.

07.05(hétfő)

Mivel az előző héten nagyrészt egy másik projekt számára végeztem kutatást, a mai napot felzárkózásra használtam, azonkívül megbeszélés után belőttük még mennyi időnk van hátra, és ennyi idő alatt mennyit tudunk elkészíteni a projektből, és az erőforrásainkat ezek szerint megpróbáltuk beosztani, több kevesebb sikerrel.

08.06(kedd)

A nap a kapott adatbázis feldolgozásával telt el.

A file-k egy háromdimenziós (idő,id,mért érték) szerkezetben tároltak szenzorméréseket,időjárásról egységnyi időközönként.

Mivel a .csv file-ban kétdimenzióban, idő függvényében ömlesztve voltak az adatok, ehhez írtam programot, mely ezeket "2,5 dimenziósra" alakítja át, azaz külön sorba írha ki id-idő függvényében az oda tartozó datát.

Délután egy rövid scram után a projekt fel lett osztva szenzorszimulátor-fog(datafeldolgozás)-és frontend részre, melynek én a szimulátor részét vállaltam el.Mivel a projekt erőforrásai szűkösek, alternatívának a SAP felület mellé javasoltam a lokális adatmegosztást hálózaton keresztül.

.08.07(szerda)

Bugfixeltem a parsert, és átalakítottam a kimenetet a SAP-HANA rendszerbe való feltöltésre alkamassá.Emellett utána néztem a Javából szerver hostolásnak, ellenben egy szimpla java szerver létrehozásán kívül nem volt több idő foglalkozni a

feladattal, emellett mivel az SAP HANA volt a kítűzött felület, nem tulajdonítottam túl nagy prioritást neki. A maradék két napban próbáltam optimalizálni a programom, és némi büszkeséggel jelentem ki, hogy az én általam készített parser gyorsabban működik mint a mások által készítettek, mivel az enyém egyetlen ciklusban képes beolvasni és leparsolni az egész file-t, és nem szükséges még egy ciklust beágyaznia. Ennek hátránya az, hogy későbbi módosítások esetén jóval nehezebb lesz hozzáigazítani a kódot a csv file-hoz mint más esetekben.

Utolsó két hét

A gyakorlat utolsó két hete eltérő volt a gyakorlat többi részétől, home-officeban kellett elvégezni. Újabb funkciók már nem kerültek hozzáadásra, cserébe átnéztük a teljes forráskódot és a külalakján javítottunk amennyit csak tudtunk. Kivettük a felesleges részeket valamint kommenteket helyeztünk el ott ahol szükségesnek éreztük - ám túlzásba nem vittük, igyekeztünk úgy kódolni hogy alapvetően könnyen olvasható legyen. Az utolsó két alatt készítettük el az 1 oldalas összefoglalót az ELTE-Soft számára, illetve ezt az egyébként folyamatosan írt haladási naplót is átnéztük és kiegészítettük. Összességében a szakmai gyakorlatunkról elmondhatjuk hogy jól megtanított az önálló munkára, hiszen nagyon sok problémával egyedül kellett szembenéznünk, ugyanakkor paradox módon a csapatmunkára is megtanított, annak ellenére hogy nem lettünk igazából csapatokra osztva mégis voltak olyan részei a munkának amit csapatban lehetett igazán jól elvégezni, aminek köszönhetően alakultak is ki kisebb-nagyobb kooperációk.