



Két hét

i.sz. 2019. Július 12.

Kiss János

ELTE Szakmai gyakorlat

Áttekintés

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

I. 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, valamint 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";
}

```

II. 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:

<https://angular.io/guide/build#proxying-to-a-backend-server>

Angular.json ban:

```

"serve": {
  "builder": "@angular-devkit/build-angular:dev-server",
  "options": {
    "browserTarget": "sap-hana:build",
    "proxyConfig": "proxy.conf.js",
    "port": 4200,
    "host": "0.0.0.0"
  }
}

```

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 összvágyon: 373608 (1e6 HUF) társaságdb: 52

