

Fejlesztési tapasztalatok

Kiss Alex
FWCO0F

július 2-15.

1. Bevezető

A szakmai gyakorlat első két hetére ütemezett feladat egy pilot projekt elkészítése volt, mely során reprodukálnunk kellett az [Országileltár](#) egy weboldalát, egy **SAP-HANA** adatbázis használatával, valamint tetszőleges front-end megoldással.

Az alkalmazás elkészítéséhez létre kellett hoznunk az adatbázist az [SAP webes vezérlőfelületén](#) egy, az Országileltár oldaláról kinyert *.csv* állományból, majd azon **adattisztítást** és **normalizálást** kellett végeznünk, továbbá egy **kalkulációs nézetet** kellett létrehozunk.

A front-end megoldást az adatbázissal az **OData** protokoll segítségével kötöttem össze, mely jelentősen megkönnyítette a back-end oldali feladatokat.

Front-end megoldásként az **Angular** környezetet választottam, és az OData segítségével az adatbázisból kinyert adatokat a minta weboldalhoz hasonlóan táblázatokban helyeztem el, adatkötés segítségével, továbbá egy grafikont jeleltettem meg, a [canvasJS](#) könyvtár segítségével.

2. Fejlesztés

2.1. Back-end

A back-end megoldásként szolgáló SAP HANA környezetet először a hivatalos oldalon, egy ingyenes fiókkal vettem birtokba, ahol [ezt](#) a tutorialt követve létrehoztam egy **Multi-Target Application**-t. Ennek során megismerkedtem az ehhez kapcsolódó alapfogalmakkal, mint például a **HDI Module** valamint **OData**-val.

2.1.1. Az adatbázis létrehozása

A tutorial keretében létrehoztam egy adatbázist is, melyet az Országileltár oldaláról gyűjtött adatokkal töltöttem fel, miután definiáltam az illeszkedő sémát, egy **HDB CDS Artifact** létrehozásával, az alábbihoz hasonló, SQL-ből megszokott módon:

```
context Tarsasagok {
    entity Tarsasag {
        TARS_ROV_NEV      : String(200);
        TARS_HOSZ_NEV     : String(200);
        CIM_EGYBEN        : String(200);
        ...
    };
};
```

2.1.2. Az adatok tisztítása és normalizációja

Miután sikerült importálni az adatokat, a sémát először is finomhangolni kellett, hogy a típusok megfeleljenek az importálandó adatoknak. Ezt követően egyes előfordulásokat törölni vagy korrigálni kellett az adatok közül, mivel azok vagy hiányosak voltak, vagy minimális eltéréssel szerepeltek a hozzá hasonló előfordulásokhoz képest.

Ezt követően a kalkulációs nézet elkészítését megalapozva, a táblát normalizálnom kellett. Ehhez, (mint később kiderült) először ki kellett egészítenem a sémákat tartalmazó fájlt újabb *Entity*-kkel, az alábbihoz hasonló módon:

```
context Tarsasagok {
  entity Tarsasag {
    ...
  };
  ...
  entity Megye{
    MEGYE_NEV : String(50);
    MEGYE_KOD : Integer;
  };
  ...
};
```

valamint létre kellett hoznom egy "kompakt" táblázatot, mely csak a kódokat tartalmazza, az ahhoz tartozó további értékeket nem. Ezután a megfelelő SQL parancsokkal feltöltöttem az újonnan létrehozott táblákat a megfelelő adatokkal.

2.1.3. Kalkulációs nézetek létrehozása

A kalkulációs nézet létrehozásához [ezeket](#) a tutorialokat olvastam át. Fontos volt, hogy a **hdiconfig** fájlban a verziószámot átírjam 2.0.30.0-ra, mivel enélkül a grafikus szerkesztő nem engedélyezett számos funkciót.

A nézet létrehozásakor habár rendelkezésre álltak a *Cube*, valamint *Dimension* opciók, az *SQL Access Only*-t választottam.

2.1.4. Az OData konfigurációja

Az **OData** modul létrehozásához [ezt](#) a tutorialt olvastam át, mely során létrehoztam egy **Node.JS** modult, majd a **server.js** fájlban a *redirectUrl* paramétert, az **.xsodata** fájlban a lekérdezendő táblák nevét módosítottam, valamint az **mta.yaml** fájlt kiegészítettem az OData modul függőségével.

2.2. Front-end

2.2.1. Az Angular környezet beállítása

Vasicsek Gábor segítségével sikerült létrehozni egy Angular projektet, viszont ahhoz, hogy el tudjuk érni az adatbázist, módosítani kellett a proxy beállítá-

sokat, mivel a szerver ekkor még nem rendelkezett a megfelelő tanúsítvánnyal. Később egy új projektet készítettem, majd a proxy beállítások módosításához [ezt](#) a tutorialt használtam.

2.2.2. Komponensek és szolgáltatások létrehozása

Ahhoz, hogy az adatbázisból kinyert adatokat megjelenítsem, a minta oldalhoz hasonlóan táblázatokat, valamint egy grafikont használtam.

Az **Angular** környezettel való ismerkedéshez [ezt](#) a hivatalos tutorialt követtem. A táblázatoknak valamint a grafikonnak külön komponenst és szolgáltatást hoztam létre, az alábbi parancsokkal:

```
ng g c <komponensnév>
ng g s <szolgáltatásnév>
```

és ezeket a **htmlmodule** nevű modullal sikerült összekötni a korábban létrehozott adatbázissal, az OData modulon keresztül.

Az OData modul *getData* függvénye paraméterként kapja meg egy táblára vonatkozó lekérdezés további paramétereit, mint például a *filter* vagy *orderby*, az alábbi módon:

```
export class OdataService {
  constructor(private http: HttpClient) { }
  configUrl = "/hana/odata.xsodata/";
  getData(params) {
    return this.http.get(this.configUrl + params);
  }
}
```

A megjelenítendő adatokat a táblázatban a megfelelő lekérdezéseket követően adatkötéssel jelenítettem meg, és az adott sorok *onClick* metódusát úgy írtam meg, hogy az lefuttat egy újabb lekérdezést, a kiválasztott régióra / megyére szűkítve, majd a soron következő táblázat(ok)at frissíti.

```
selectRegion(val): void {
  this.odata
    .getData([
      "proba?$select=MEGYE,JEGYZ_TOKE_ERT_HUF&$$filter=REGIO eq '"
      + val + "'"
    ])
    .subscribe((res: any) => {
      this.countyData = res.d.results;
    });
  ...
}
```

A grafikonhoz a **canvasJS** könyvtárát használtam, melyhez a szükséges adatokat szintén az OData modullal szereztem meg, egy *select-box*-ban kiválasztható évszámokkal való szűkítéssel. A select-box *onChange* metódusához az alábbi kódot kötöttem hozzá, mellyel frissül a grafikon:

```
onChange(val) {
    this.year = val;
    this.odata
        .getData([
            "Tarsasag?$top=1000&$orderby=TARS_ROV_NEV&$filter=ASZ_EVE eq '"
            + val + "'"
        ])
        .subscribe((res: any) => {
            this.data = res.d.results;
        });
    ...
    this.chartService.show_graph(db);
}
```

Habár a minta oldalon a select-box fix adatokkal dolgozik, én az *<option>* mezőit az adatbázisban található évszámok alapján az OData rétegen keresztül töltöttem fel.