

Dokumentáció

07. 01.

Kick-off esemény végighallgatása, érdeklődési körök felvázolása, kezdetleges csoportfelosztás, ismerkedés az SAP Cloud Foundry-val.

07. 02.

Önálló munka kezdete, szükséges ismeretek elsajátításának megkezdése.

developers.sap.com oldal böngészése, ez alapján próbálkozás Node.js modul létrehozásával a jövőbeni front end kialakításához. Akadályba ütközés bizonyos funkciók hiánya miatt az általunk alkalmazott platformon. Ugyanerre vezetett egy HTML5 modul létrehozására való kísérlet.

Félidőben témavezetői segítséggel első táblázat létrehozása SAP HANA-ban. Próbálkozás „Hello World” alkalmazás létrehozásával, szerverproblémák miatt sikertelenül.

Maradék időben informálódás Node.js-ről.

07. 03.

JavaScript nyelv tanulása további instrukciók megadásáig.

9:40 – mail érkezése elsajátítandó anyagokról.

9: 52 – tanári tájékoztató a legfontosabb témákról, a gyakorlat időbeosztásáról.

A nap hátralévő részében az ajánlott linkek olvasása és jegyzetelése. Ezek a következők:

- XSA dokumentáció:
<https://help.sap.com/viewer/4505d0bdaf4948449b7f7379d24d0f0d/2.0.03/en-US/d8226e641a124b629b0e8f7c111cd1ae.html>
- CDS dokumentáció:
<https://help.sap.com/viewer/09b6623836854766b682356393c6c416/2.0.02/en-US/0b1eb07d74ec4f91947ff4cc4f557429.html>
- PAL dokumentáció:
<https://help.sap.com/viewer/2cfbc5cf2bc14f028cfbe2a2bba60a50/2.0.03/en-US/c9eed704f3f4ec39441434db8a874ad.html>

07. 04.

Dokumentációk feldolgozásának folytatása.

11:14 – tanári oldalra csv fájl felkerülése, információk meghallgatása a teendőkkel kapcsolatban. Szerverhiba miatt a feltöltés nem működik az SAP HANA platformon, így az adatok tisztítását kezdjük meg a csv fájlban.

07. 05.

Szerverhibák okainak felmérése témavezetői segítséggel, a megoldásig informálódás az OData-ról és a front end lehetőségeiről.

12:25 – saját szerver elkészül, ezen már működik az adatbázis, elkezdődhet a táblázat létrehozása és a csv fájl importálása. Amint megvan, a maradék időben további adattisztítással foglalkozunk.

Az adatok feltöltésének lépései:

- Workspace-be „New Project from Template” (nálam a *nyarigyak* nevet kapja)
- *nyarigyak*-ba „New SAP HANA Database Module” (*nyaridb* nevet kapja)
- *nyaridb src* mappájába „New HDB CDS Artifact” (*tablazat.hdbcds*)
- a kontextus nevét a vizuális szerkesztőben „*tablazat*”-ra állítottam, majd dupla kattintással belementem
- a kontextuson belül létrehoztam a „*database*” entitást, ez fogja tárolni az összes adatot
- dupla kattintás a „*database*”-re megnyitja az adattagok hozzáadásának lehetőségét; itt manuálisan felvettem a csv fájl fejléceiben megadott mezőket, beállítottam a mezőtípusokat
- a *nyarigyak*, majd a *nyaridb* sikeres buildelése után átmentem a Database Explorerbe
- itt rákattintottam az „Add a database to the Database Explorer” (plusz ikonnal jelölt) lehetőségre, és a felugró listából kiválasztottam a *nyarigyak*-ot
- ezt követően a bal menüsávban megjelent az imént létrehozott adatbázis, a „Tables” fülben megtaláltam a „*database*”-t
- a „*database*”-re jobb gombbal kattintva megjelent az „Import Data” lehetőség, amire kattintva egy felugró menüben megadtam a csv fájl forrásnak
- néhány tulajdonságot az SAP-s táblázat és a csv között manuálisan kellett beállítani, ezek után az importálás sikeres volt
- további adattisztítás szükségessége miatt a Database Explorer konzolablakát megnyitva, SQL parancsok segítségével készítettem egy *MODOSITOTT* nevű táblát, melybe átmásoltam a *database* adatait, majd ezeket update-eltem (2 sor törlésre került hibás, illetve teljesen hiányzó adatok miatt)

Első hét konklúziója: szerverhibák miatt az első hét javarészt az elméleti alapok elsajátításáról szólt; péntek délutánra a saját szerveren mindenki elkészítette a saját táblázatát, importálta az adatokat.

07. 08.

Az első órában az adatok megjelenítésének csoportos tervezése. Általános megegyezés az adatok szűrésére év és/vagy megye/régió alapján.

Következő lépés az OData létrehozása; ehhez először anyagok olvasása (developers.sap.com, help.sap.com). Hiábavaló próbálkozás az olvasottak reprodukálására.

14:12 – tanári megbeszélés során kiderült, hogy az OData jogosultsági problémák miatt nem működik. Ezt követően az előző heti munkát foglaltuk össze szóban.

07. 09.

Adatok további tisztítása.

10:23 – témavezetői segítséggel sikerül létrehozni az ODatát. Ennek lépései:

- *nyarigyak*-ba „New Node.js Module” (fontos az „Enable XSJS support” kipipálása, a modult *odata*-nak neveztem el)
- *odata lib* mappájába „New File” (*myodata.xsodata*), ebbe a következő kód került:

```
service{  
    "MODOSITOTT" as "DATABASE";  
}
```
- *server.js* fájl 11. sorában a *redirectUrl*-t átírtam „*/myodata.xsodata*”-ra
- build után kaptunk egy linket (<https://oktnb132.inf.elte.hu:51045/myodata.xsodata>), ennek végére különböző lekérdezéseket illesztve kapjuk meg a kívánt adatokat (pl. [https://oktnb132.inf.elte.hu:51045/myodata.xsodata/\\$metadata](https://oktnb132.inf.elte.hu:51045/myodata.xsodata/$metadata))

11:30 – vissza az önálló munkához. Altáblázatok létrehozása, melyek a kalkulációs nézethez fognak kelleni. Ezek „kereteit” a *nyarigyak/nyaridb/src/tablazat.hdbcds* fájl vizuális nézetében csinálom meg, csakúgy, mint egykor a *database-t*. Mikor végeztem a létrehozással, a Database Explorer konzolablakában, SQL utasításokkal töltöttem fel az altáblázatokat a megfelelő adatokkal.

A következő altáblázatokat hoztam létre:

- TARSKODOK(TARS_ROV_NEV, TARS_HOSZ_NEV, CIM_EGYBEN, TARS_TIPUS_KOD, ADOSZAM, GAZD_FORM_KOD, CEGALL_KOD, NEMGAZD_AG_KOD, NEMGAZD_AGAZAT_KOD, NEMGAZD_SZAKAGAZAT_KOD, JEGYZ_TOKE_ERT_HUF, ORSZAG_KOD, REGIO_KOD, MEGYE_KOD, TELEPULES_KOD, ASZ_EVE)
- TARSTIPUS(TARS_TIPUS_MEGNEV, TARS_TIPUS_KOD)
- GAZD(GAZD_FORM, GAZD_FORM_KOD)
- CEGALL(CEGALL, CEGALL_KOD)
- NEMAG(NEMGAZD_AG_MEGNEV, NEMGAZD_AG_KOD)
- NEMAGAZAT(NEMGAZD_AGAZAT_MEGNEV, NEMGAZD_AGAZAT_KOD)
- NEMSZAKAGAZAT(NEMGAZD_SZAKAGAZAT_MEGNEV, NEMGAZD_SZAKAGAZAT_KOD)
- ORSZAG(ORSZAG, ORSZAG_KOD)
- REGIO(REGIO, REGIO_KOD)
- MEGYE(MEGYE, MEGYE_KOD)
- TELEPULES(TELEPULES, TELEPULES_KOD)

07. 10.

Olvasás OData felhasználási lehetőségeiről.

10:00 – témavezető segítségével Angular telepítése, de nem működik a gépeken.

A következő órák azzal telnek, hogy a témavezető és az egyik hallgató kolléga működőképessé tegye az Angulart, közben mindenki más önálló munkát végez.

A nap végére az Angular-project sikerrel járt, kör-emailben kaptunk egy tömörített mappát, amely keretül szolgálhat saját Angularos kódunk megírására.

07. 11.

Calculation View létrehozása egy hallgató kolléga segítségével. Lépései:

- *nyarigyak src* mappájának *.hdbconfig* fájljában *plugin_version* átírása 2.0.30.0-ra (csak ezzel működik a kalkulációs nézet)
- szintén *src* mappába „New Calculation View” (*Calculation.hdbcalculationview*)
- Create Join-nal behúzok egy joint a szerkesztőbe, amit „data”-ra nevezek át
- „Add Data Source” funkcióra kattintás után megadom az altáblázatokat forrásként
- ezután dupla kattintás a „data”-ra – Join Definition fülben az inner joinok meghatározása az adattagok összekötésével
- ezt követően a Mapping fülbe az adattagok behúzása (mindegyiket csak egyszer, tehát mivel a MEGYE-ben szerepel a MEGYE és a MEGYE_KOD, a TARSKODOK-ból már nem húzom be a MEGYE_KOD-ot még egyszer)
- szerkesztő bezárása, dupla kattintás az Aggregationre
- Mapping fülben itt is minden adattagot behúzunk
- build után működik a Data Preview

A maradék időben az előző nap kapott Angularos kód szerkesztése úgy, hogy az saját OData-val működjön; a munkaidő végéig sikertelen.

07. 12.

További munkák az Angularos kódon, félidőben sikerült a keretet saját OData-ra futtatni. A maradék idő a megjelenítés szerkesztésével telt (cím átírása, felesleges szövegek törlése, több szűrő megvalósítása).

Alább felsorolom a „keretkód” módosításait.

odata.service.ts:

configUrl-t át kellett írni saját odatára: configUrl = '/hana/myodata.xsodata/';

proxy.conf.js:

Ebben a fájlban adjuk meg, hogy az előző configUrl elején mi az a „hana”. Ehhez a targetet kellett módosítani: target: "https://oktnb132.inf.elte.hu:51045",

proxy.conf.json:

Itt is át kellett írni a targetet sajátára: target: "https://oktnb132.inf.elte.hu:51045",

app.component.html:

Ebben a fájlban írjuk le a megjelenítendő oldal kinézetét. Egy üdvözlő címsor után az oldalon kirajzolódik egy diagram, melyen a cégek bevételei láthatóak. A megjelenített adatokat évek és régiók szerint szűrhetjük (a két szűrő egyszerre érvényesül). A szűréshez a listaelemeket odata segítségével kérdezzük le, melyet a következő fájlban állítunk be.

app.component.ts:

A megjelenítés háttérlogikája. Az ngOnInit függvény beállítja a szűrők kezdőadatait, lekérdezi az odatából a listaelemeket. A show_graph függvény kirajzolja a grafikont, az onChange pedig a grafikonon megjelenő adatokat módosítja a beállított szűrők függvényében. Ebben a fájlban szükség volt a kalkulációs nézet lekérdezéseire is, így az SAP Web IDE-ben lévő myodata.xsodata fájlt ki kellett egészíteni:

```
service{  
  "MODOSITOTT" as "DATABASE";  
  "Calculation" as "kalkulacio";  
}
```

Második hét konklúziója: a második hét végére rendelkezésre áll egy, a csv fájlból importált adatbázis SAP HANA-ban, ehhez tartozó altáblázatok, kalkulációs nézet és OData, illetve az ODatát felhasználó, még kezdetlegesnek mondható megjelenítés. A hétvége feladata a dokumentáció elkészítése, illetve a továbbhaladás átgondolása, célkitűzés a következő 4 hétre.