

# Haladó web alapú programfejlesztés

## Beadandó feladat

A következő feladatpontokhoz készítsen Angular alapú alkalmazást felhasználói felületekkel. A megvalósítás során alkalmazza a kurzuson bemutatott elveket, technikákat, eszközöket. A szerver műveleteket valamilyen mock-olt megoldással legyenek megvalósítva (InMemoryWebApi, Http interceptor-okból adott válaszok, egyéb létező mock megoldás). Az értékelés során figyelembe lesz véve a modulok/komponensek, tervezési minták, szoftverfejlesztési alapelvek megfelelő használata, illetve az RxJS Store használata az adatok/állapot menedzselésére.

1. Készítsen alkalmazást, amely vasúti személykocsikat képes nyilvántartani. A vasúti személykocsiknak van sorozatjele (PI: Bhv, BDbhv, AcBc stb.), gyártási éve, egyedi pályaszám (PI: 50 55 20-05 555-7, formázás/white space-ek nélkül legyen tárolva nélkül) tulajdonosa (PI: MÁV), telephelye (PI: Celldömölk). A telephelyeknek van neve, tulajdonosa (PI: MÁV), postai címe és egyedi kódszáma (PI: Celldömölk kódszáma 02170).
2. Készítsen felhasználói felületet, amely lekérdezi és megjeleníti táblázatosan az összes (nem törölt) vasúti kocsit. A táblázatban csak a sorozatjel, pályaszám és telephely jelenjen meg. A felületen legyen egy gomb, amellyel új kocsit lehet felvinni (annak minden adatával) egy megnyíló ablakban. Az összes vasúti kocsi listázásánál lehessen megadni, hogy a töröltek is legyenek a listában vagy ne. A táblázat minden sorában legyenek akciógombok:
  - a. Lehessen szerkeszteni a kocsi adatait (itt már az összeset)
  - b. Lehessen törölni az adott kocsit.
3. Az alkalmazásban a kocsi listázó táblázatban lehessen lekérni egy telephelyhez tartozó összes kocsit egy telephelyre kattintva (a 2. pontban elkészített listázó komponens legyen használva).
4. Készítsen felületet, amely lekéri és megjeleníti a rendszerben tárolt telephelyek listáját, annak minden adatával. Ezen a felületen lehessen új telephelyet felvinni a rendszerbe, a kocsi listázós felülethez hasonlóan. Minden telephely sorhoz tartozzanak a következő akciógombok:
  - a. Telephely adatainak szerkesztése
  - b. Telephely logikai törlése (ellenőrizve, hogy ha a telephelyhez van kocsi rendelve, akkor ne lehessen törölni a telephelyet)
  - c. Telephelyhez rendelt összes kocsi megjelenítése (a 3. pontban elkészített komponens segítségével)
  - d. A logikailag törölt telephely visszaállítása
5. A 2. pontban elkészített hozzáadás és szerkesztés műveleteket módosítsa úgy, hogy a kocsi telephelyét egy legördülő listából lehessen kiválasztani a rendszerben található (nem törölt) telephelyek közül.
6. A 2. és 3. pontban elkészített felületekhez készítsen szűrő sávot, amelyben a kocsi/telephely minden adatára lehet szűrni. A szűrési feltételek között ÉS kapcsolat legyen. A szűréseket egy közös komponensben valósítsa meg megfelelő paraméterezést használva.
7. A 2. és 3. pontban elkészített felületek mindegyikén lehessen az adatokat sorba rendezni a tulajdonságok bármelyike alapján (egyszerre elég 1 tulajdonság alapján rendezni a listát).
8. Minden űrlapon ahol felvinni vagy módosítani lehet a rekordokat legyenek a mezőkhöz definiálva megfelelő validációs szabályok (az egyes paraméterek nincsenek meghatározva, értelmes értékek legyenek választva pl. bevihető karakterek maximális számához, stb.) az alábbiak figyelembe vételével:

Beadandó feladat

- a. Az egyedi adatok (pályaszám, kódszám) egyedisége legyen ellenőrizve aszinkron validátorral és legyen jelölve a felületen, amíg tart az ellenőrzés,
- b. A pályaszámok ellenőrzése legyen kiegészítve egyedi validációs szabállyal az alábbiaknak megfelelően: A pályaszám utolsó számjegye egy önellenőrző szám, ez alapján a szám alapján lehet validálni.

Ezt az előző 11 jegyből számítják ki a következőképpen: a páratlan helyen álló számokat 2-vel szorozzák, a páros helyen állókat 1-gyel, és mindegyik szorzatban veszik a számjegyek összegét; majd az így kapott 11 számot összeadják, végül a 12. jegyet úgy választják meg, hogy az előbbi összeghez hozzáadva 10-zel osztható számot kapjanak. Pld, ha az 51553930016 tizenegy-jegyű számból indulunk ki:

a 11 jegyű szám:	5	1	5	5	3	9	3	0	0	1	6
szorzó:	2	1	2	1	2	1	2	1	2	1	2
szorzat	1	1	1	5	6	9	6	0	0	1	1
	0		0								2
a jegyek összege:	1	1	1	5	6	9	6	0	0	1	3

akkor az összeg:  $1+1+1+5+6+9+6+0+0+1+3=33$ , tehát az utolsó, 12. jegy a 7 lesz. A vasúti kocsi 12 jegyű száma: 515539300167.

9. A kocsik és a telephelyek kezelése külön-külön feature modulokba legyen kiszervezve (lazy loading + navigáció).
10. Készítsen egy `HttpInterceptor`-t, ami minden kéréssel elküld egy `X-Request-Id` headert, ami egy random generált GUID-ot tartalmaz.
11. Készítsen egy `pipe`-ot ami a kocsi pályaszámok formázását végzi el 50 55 20-05 555-7 formátumban, alkalmazza a `pipe`-ot azon helyeken ahol megjelenik ez az adat.