

Fejlesztési feladat

Tartalom

1. Szcenárió ismertetése.....	1
2. Probléma ismertetése.....	1
2.1. Kapcsolattartó entitás.....	2
2.2. Listázás folyamata.....	2
2.3. Létrehozás folyamata.....	2
3. Feladat.....	3

1. Szcenárió ismertetése

Adott egy Angular alkalmazás és egy hozzá tartozó, Spring Boot-ban készült backend service, ami REST API-t szolgáltat az alkalmazás számára.

2. Probléma ismertetése

Az ügyfél igény az, hogy az alkalmazást kiegészítsük új felületekkel, ahol kapcsolattartók kezelésére van lehetőség. Egy új menüpont alól elérhető a listázó felület, ahonnan további műveleteket tud kezdeményezni a felhasználó.

Szükséges műveletek listája:

- Kapcsolattartók listázása
- Kapcsolattartó létrehozása

A kapcsolattartókat a rendszer a hozzá tartozó adatbázisban tárolja el.

2.1. Kapcsolattartó entitás

Az ügyfél által kezelni kívánt kapcsolattartói adatokat az alábbi táblázat tartalmazza.

Adat neve	Típus	Leírás
Vezetéknév	szöveg	
Keresztnév	szöveg	
E-mail cím	szöveg	Csak e-mail cím formátum lehet
Telefonszám	szöveg	Opcionális; Csak telefonszám formátum lehet a következő mintának megfelelően: +36 ## ### ####
Cég	hozzárendelés	Kiválasztható egy darab cég az adatbázisból
Megjegyzés	szöveg	Opcionális; Tetszőleges megjegyzést tartalmazhat
Státusz	enumeráció	Választható: aktív, törölt

táblázat 1 - kapcsolattartó adatai

Fejlesztői oldalról egészítsük ki a fenti adatokat további adatokkal:

- egyedi, generált azonosító
- létrehozás ideje
- utolsó módosítás ideje

2.2. Listázás folyamata

A felhasználó egy listát szeretne látni a rendszerben lévő aktív kapcsolattartókról. Ennek folyamata:

1. A felhasználó kiválasztja a listázó menüpontot
2. Az alkalmazás lekérdezi az első oldalhoz tartozó kapcsolattartók listáját. Az oldalon megjelenik egy listázó táblázat, az alábbi oszlopokkal:
 - a. Teljes név
 - b. Cég neve (hozzárendelt céghez tartozó név)
 - c. E-mail cím
 - d. Telefonszám
3. A táblázat csak az aktív státuszú kapcsolattartókat jeleníti meg, név szerint ABC sorrendben és egyszerre mindig tízet.
4. A felhasználónak lehetősége van tízesével előre és hátra lapozni a találatok között. Ilyenkor az alkalmazás újabb lekérdezéseket hajt végre a háttérrendszerhez, ahol a kérés paraméterként tartalmazza a megjeleníteni kívánt oldalszámot, válaszként pedig letöltésre kerül a következő tíz elem.

2.3. Létrehozás folyamata

A felhasználó létre szeretne hozni egy új kapcsolattartót. Ennek folyamata:

1. A felhasználó megnyitja a listázó oldalt
2. Rákattint a listázó felett elhelyezkedő Létrehozás gombra
3. Az alkalmazás megjelenít egy modalt, benne egy létrehozás űrlappal ami az alábbi mezőket tartalmazza:
 - a. Vezetéknév, szöveges beviteli mező
 - b. Keresztnév, szöveges beviteli mező
 - c. Cég, kiválasztás legördülő menüvel
 - d. E-mail cím, szöveges beviteli mező

- e. Telefonszám, szöveges beviteli mező
 - f. Megjegyzés, szöveges beviteli mező, többsoros
4. Az űrlap alatt elhelyezkedő mentés gomb megnyomását követően az alkalmazás kérésben elküldi az űrlap tartalmát. A mentés gomb csak akkor kattintható, ha a megadott adatok megfelelnek az alábbi validációs szabályoknak:
 - a. Minden nem opcionális mezőnél van megadott adat
 - b. A telefonszám csak telefonszám formátumú lehet (lásd **táblázat 1 - kapcsolattartó adatai**)
 - c. Az e-mail cím csak e-mail cím formátumú lehet
5. Az elküldést követően a háttérrendszer ellenőrzi a felküldött adatokat és csak abban az esetben hozza létre a kapcsolattartót, ha sikeres volt az ellenőrzés. A validációs szabályok a következők:
 - a. Minden nem opcionális mezőhöz érkeznie kell adatnak
 - b. E-mail cím csak e-mail cím formátumú szöveg lehet
 - c. Az e-mail cím domain része nem szerepelhet az előre meghatározott tiltólistán
 - d. A telefonszám csak telefonszám formátumú lehet (lásd **táblázat 1 - kapcsolattartó adatai**)
 - e. Az érkező cég azonosítónak létező céghez kell tartoznia
6. Siker esetén létrejön egy aktív státuszú kapcsolattartó, a modal pedig bezárásra kerül. A listázó által megjelenített adatok frissítésre kerülnek, hogy látható legyen az újonnan felvett elem.
7. Hibás felküldés esetén a rendszer sikertelen státuskóddal válaszol a kérésre. A validációs hibákat a kliens alkalmazás megjeleníti, a form pedig legyen újra beküldhető.

3. Feladat

A feladat, hogy készíts megvalósítást az ismertetett folyamatokhoz az alkalmazás backend és frontend oldalán is. A megvalósításnak REST API szeretnénk, a kommunikáció formátumának JSON-t. Minden felhasználói eset egy végpontnak megfeleltethető.

Hitelesítéssel, jogosultságkezeléssel nem foglalkozunk.

Kérések és megjegyzések a megvalósítással kapcsolatban:

- Szolgáltatunk egy repository-t, ami tartalmaz egy Docker Compose fájlt, infrastruktúra biztosításához. Az átadott repository tartalmaz egy README.md fájlt telepítési és használati instrukciókkal. Kérjük, hogy ebbe hozz létre egy új Spring Boot és egy új Angular projektet, amibe a megoldást elkészíted.
- Az elkészült kódhoz szeretnénk, ha írnál teszteteket. A kliens esetén nem kérünk E2E vagy DOM vizsgálatot végző teszteteket.
- Az mellékelt adatbázis tartalmaz előre összeállított adattáblákat őssadatokkal a cégekhez és az e-mail domain tiltólistához, kérjük, hogy ezeket használd fel.
- A feladat megoldásához kérjük, hogy ahol csak lehet alkalmazd az integrált Angular Material könyvtár által szolgáltatott komponenseket.