Programrendszerek fejlesztés projektmunka

Gáspár Tamás

April 21, 2021

Contents

1 Linkek

Az alábbiak az alkalmazáshoz tartozó legfontosabb linkek.

- A projekt GitHub repozitóriuma

2 A fejlesztés menete

Itt részletesebben leírom, hogy egyes commitok mit adtak hozzá, vagy változtattak meg. Minden esetben megadok egy linket is a commitra, hogy ez ellenőrizhető legyen. A commitokban ezt a dokumentációt is megváltoztatom, de ezt nem írom itt le. Az olyan commitok nem kerülnek ide, ahol csak a dokumentációt frissítettem.

2.1 Alapozás

A commit teljes szövege: *NodeJS és angular hozzáadása. NodeJS kezdeti konfigurálás, Angular alap frontend.*

Ez a commit itt található.

Létrehoztam a projekt struktúráját, majd az Angularos és a NodeJS projekteket. Eldöntöttem, hogy az angular alkalmazást a NodeJS szerver fogja hostolni, és kidolgoztam ennek a módját. Letöltöttem a szükséges *npm* csomagokat, és megírtam a szerver vázát. Az angular alkalmazáshoz egyszerű frontendet készítettem, ebbe fognak majd a komponensek kerülni.

2.2 Felhasználói felület létrehozása

A commit teljes szövege: *Angular frontend fejlesztése*

Ez a commit itt található.

Megalkottam a felhasználói felületet az angularban. A főkomponens tartalmazza a menüsort, és a bejelentkezési állapotot, és ezen belül jelennek meg az egyes komponensek. A hozzáadott komponensek:

- Login: bejelentkezés
- Register: regisztráció
- Home: főoldal
- Products: termékek
- About: információ

2.3 Authentikáció

A commit teljes szövege: *Felhasználókezelés megvalósítása*

Ez a commit itt található.

Kliens és szerveroldalon is implementáltam a felhasználókezelést. Ez szerver oldalon azt jelenti, hogy létrehoztam a login, logout és user endpointokat, és az felhasználó adatbázis sémáját. Angular esetén funkcionalitás került a login és register komponensek mögé, és bekerült egy guard, ami ellenőrzi a bejelentkezést.

Még itt elég sok a hiba, a későbbiekben ezen jelentősen javítok. Pl: bejelentkezés ellenőrzése service-el, nem pedig a local storage-al, vagy a loading indicator mutatása amíg a bejelentkezés tart.

2.4 Felhasználókezelés

A commit teljes szövege: *Backend: felhasználói adatok frissítése user endpointon. UI: sweetalert, material, felhasználói felületadatmódosításra*

Ez a commit itt található.

Szerver oldalon bővítettem a */user* endpointot, hogy lehessen felhasználói jelszót változtatni, és felhasználót törölni. Kliens oldalon ehhez egy felületet hoztam létre, ahol a bejelentkezett felhasználó módosíthatja a jelszavát vagy törölheti magát.

Szépítettem az oldal kinézetét, a Google material komponensek használatával, és az alap JavaScript alert dialógus helyett egy alert könyvtárra tértem át.

2.5 Termékkezelés admin felületről

A commit teljes szövege: *Backend: product endpoint, Frontend: admin felület*

Ez a commit ... található.

Szerver oldalon létrehoztam a */product* endpointot, a termékek módosítására és lekérésére. Ehhez egy admin felületet is létrehoztam. Ezt a felületet csak bejelentkezett admin érheti el.

Itt lehetőség van új termék létrehozására, meglévő frissítésére, és törlésére. Létrehozás és frissítés úgy történik, hogy fel lehet tölteni egy JSON fájlt, ami a termék adatait tartalmazza. Egy ilyen példa fájlt beraktam a *docs* mappába is, *imaginaryPiece.json* néven.

3 API pontok

Itt dokumentálom, hogy milyen API endpointokat lehet használni. Mindegyik endpoint a */api* mögött van. Ha az URL-ben nincs */api*, azt az angular fogja lekezelni, nem pedig a szerver.

Például:

http://localhost:3080/api/status

3.1 /status

Ez csak GET-re válaszol, és egyszerűen visszaküld egy kis szöveget. Arra használható, hogy megnézzük hogy működik-e a szerver. Az endpointok a nodeJS mappa endpoints almappájában vannak definiálva, kivéve a */status*, mert az annyira egyszerű.

3.2 /login

Ide csak POST-olni lehet, és meg kell adni a felhasználónevet és jelszót. Ha ezek jók, akkor a bejelentkeztetés megtörténik.
Az adatokat JSON-ban várja:

```
{
 "username": "valaki",
 "password": "valaki_jelszava"
}
```

A választ JSONben küldi:

```
{
 "message": "Pl Sikeres bejelentkezes",
 "isAdmin": "false"
}
```

Az angular nézi az *isAdmin* értékét, hogy tudja, hogy admin jelentkezett-e be.

3.3 /logout

A */login* párja, ami kijelentkezteti a felhasználót. Csak POST-ot fogad, nem vár semmilyen adatot és akkor sem dob hibát, ha nem volt senki bejelentkezve.

3.4 /user

Felhasználókezelő endpoint, ami POST-ot, PUT-ot és DELETE-et támogat.

POST esetén regisztáció történik. JSON-ban meg kell adni a felhasználónevet és jelszót (úgy, mint a */login* esetén).

PUT esetén a jelszó frissíthető, itt meg kell adni a felhasználónevet és **jelenlegi** jelszót, majd az új jelszót:

```
{
 "username": "valaki",
 "password": "valaki_jelszava",
 "newPassword": "valaki_uj_jelszava"
}
```

Csak akkor lesz változtatás, ha a felhasználó be van jelentkezve, és a jelenlegi jelszava egyezik.

DELETE-tel felhasználó törlés kell. PUT-hoz hasonlóan itt is küldeni kell a felhasználónevet és a jelszót, csak akkor fog végrehajtódni, ha be vagyunk jelentkezve és a jelszó helyes.

3.5 /product

Ezen az endpointon kérhetőek le, vagy módosíthatóak a termékek. Mindegyik híváshoz be kell, hogy jelentkezzünk. Azokat hívásokat, amik a termékek adatbázisát megváltoztatják, csak admin hajthatja végre.
GET: Ez lekéri az összes terméket, amiket egy JSON fájlban küld vissza.
POST (admin): Új termék hozható létre vele. Várja a hitelesítő adatokat, és a termék adatait, a következő formában:

```
{
 "username": "valaki_admin",
 "password": "valaki_jelszava",
 "name": "termek neve",
 "description": "termek leirasa",
 "price": "termek ara",
 "imgPath": "utvonal a termek kepere"
}
```

PUT (admin): POST-hoz hasonló, de itt a terméknek már léteznie kell, és az adatai frissítve lesznek. Pont ugyanolyan formában várja az inputot, mint a POST.
DELETE (admin): Termék törlése. Az inputot a következő formában várja:

```
{
 "username": "valaki_admin",
 "password": "valaki_jelszava",
 "name": "torlendo termek neve",
}
```

4 Bemutató

Ahogy a követelményekben megvan, lehetőség van a belépni a következő felhasználóval:

- Username: szaboz
- Jelszó: PRF2021

Ez a felhasználó egyben admin is. Lehet regisztrálni persze más felhasználót is, az nem lesz admin.