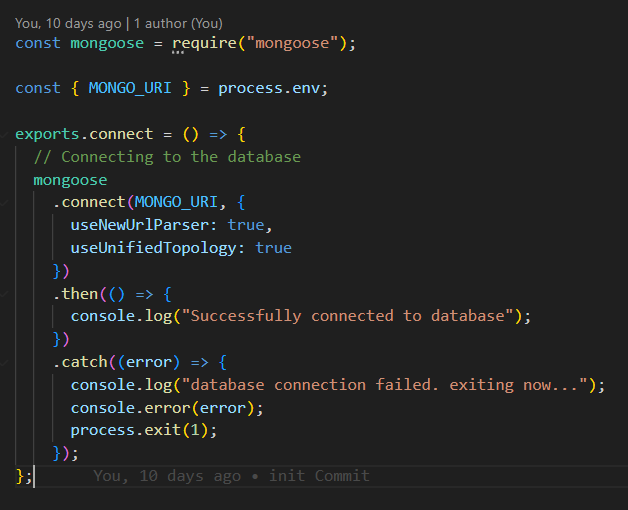
Webtech2 – Warehause-webapp

# Vékony Róbert H0F0SZ

1. Az alkalmazás
   1. Angular
   2. NodeJS
   3. MongoDB

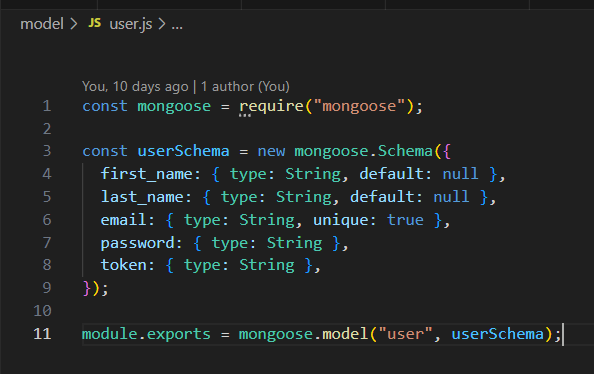
Az alkalmazásom célja egy bejelentkező felületet követően raktárkészlet feltöltése, listázása, módosítása.

Az alap angular-os webapp legenerálását követően a mongoDB kapcsolattal kezdtem.

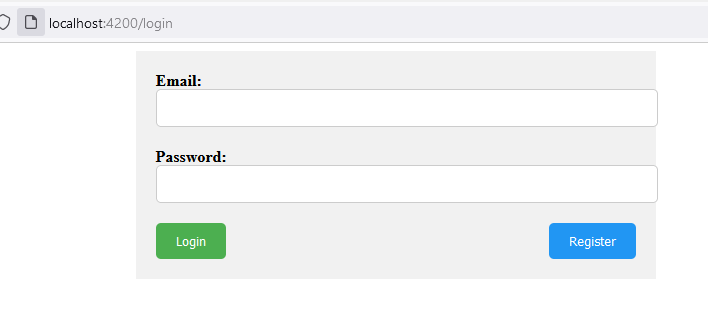


MONGO\_URI=mongodb://127.0.0.1:27017/warehauseDB

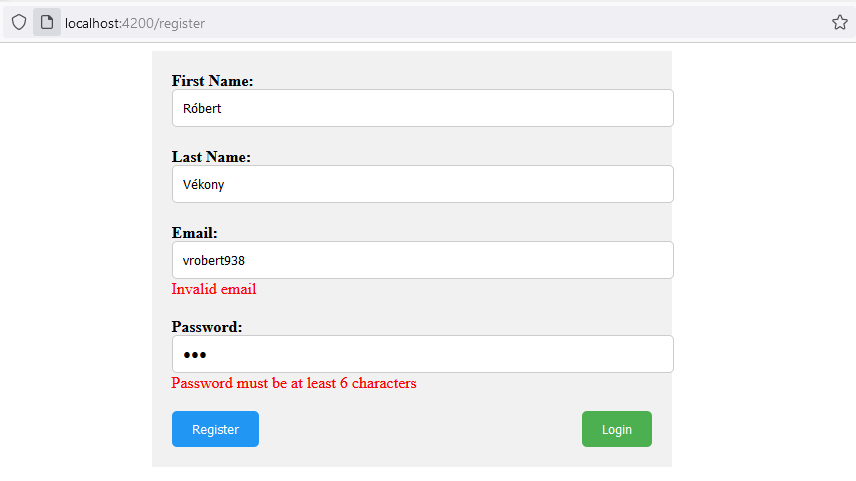
A sajátgépemen futtatom a mongoDB-t így localhoston a warehauseDB-nek elnevezett adatbázist használom. Miután sikeresen csatlakoztam a MongoDB-hez kezdődhetett a UserLogin kialakítása.

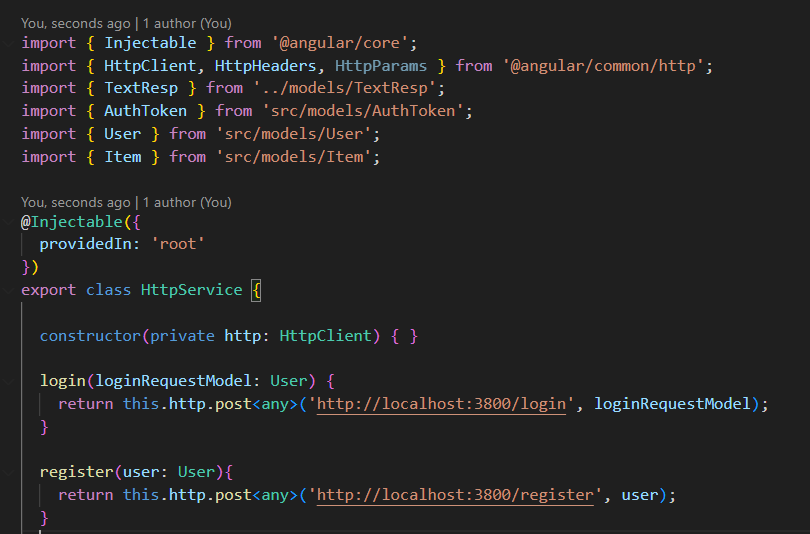
Létrehoztam a login és register component-et majd a szükséges UserSchema-t amit szeretnék letárolni az adatbázisban.   
  


A bejelentkező és a regisztrációs oldal elkészült

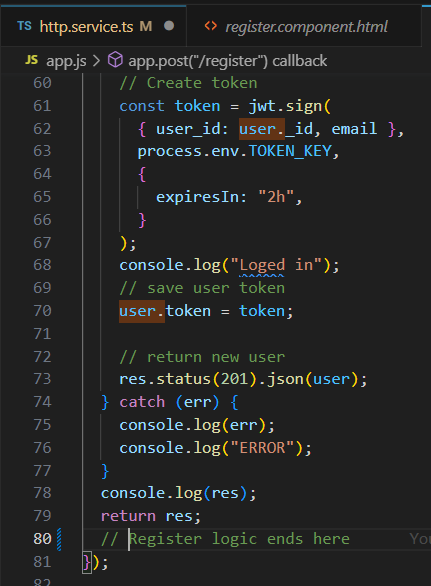


A regisztrációnál a validálást az input mezőre állított feltétellel végzem és mező alatt megjelenítem, ha nem megfelelő.   
„<div \*ngIf="form.controls.password.invalid && (form.controls.password.dirty || form.controls.password.touched)" class="error-message">Password must be at least 6 characters</div>”



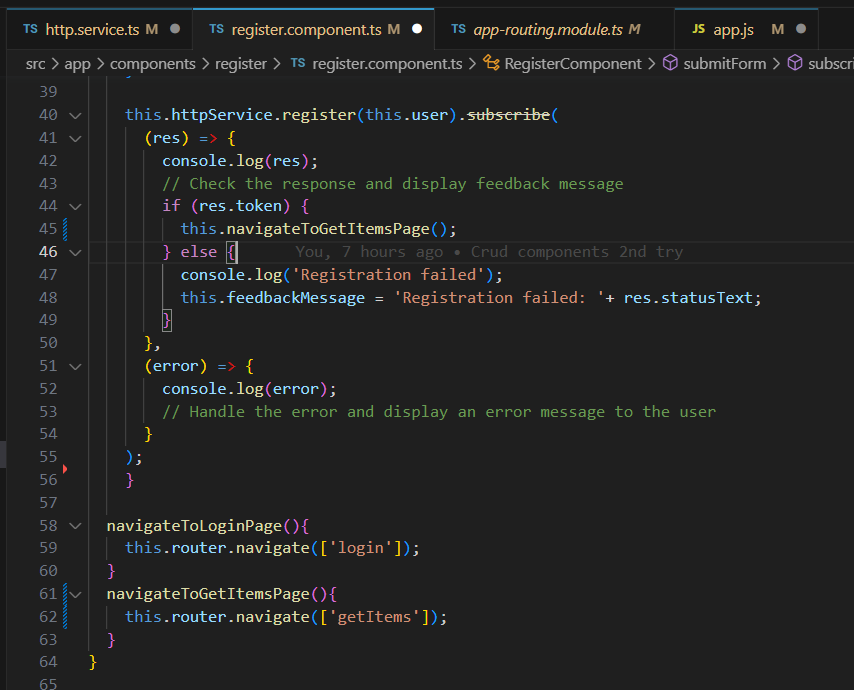
Létrehoztam a http.service.ts-ben a login és register postok hívását.  


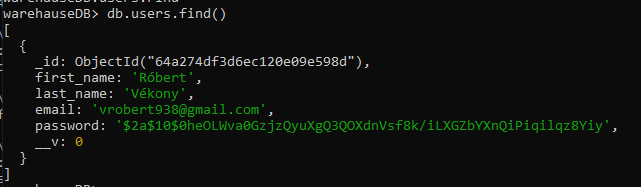
A http.service.ts-ben található register az app.js-ben lévő post metódust hívja ami request.body-ban kapja json formátumban a form tartalmát.  
  

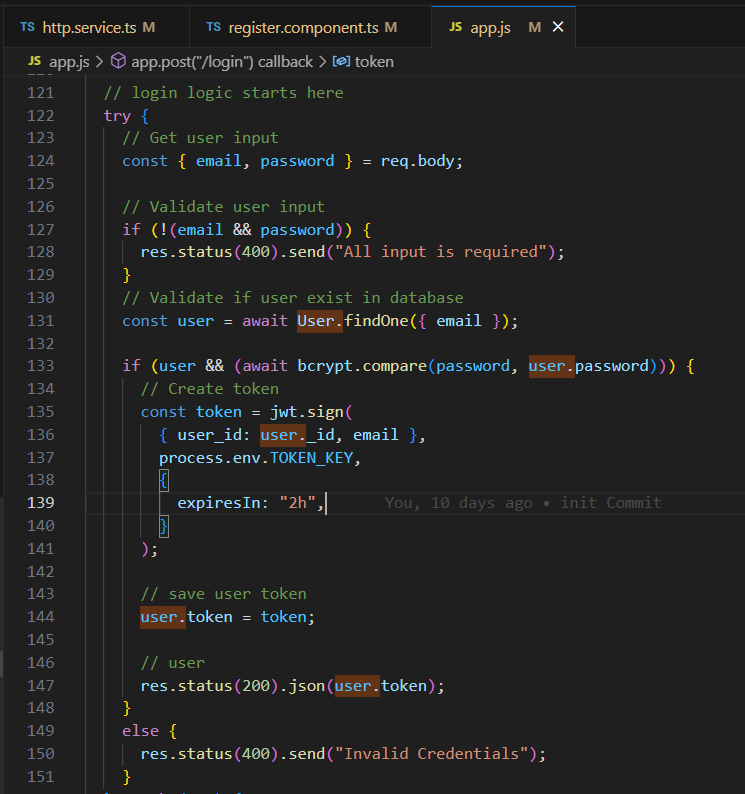

Miután ellenőriztem, hogy minden szükséges adat megérkezett a User.findOne({email})-el ellenőrzöm, hogy az adott emailcímmel, történt-e már regisztráció. Ha nem, a bcrypt-et használva a text alapú jelszóból encryptelt jelszót állítok elő.  
Ezekkel az adatokkal a User.Create(„user adatai”)-val létrehozom az adatbázisban a felhasználót.  
  


a Json web token használatával generálok egy 2 órás tokent. majd a User modelt

A sikeres regisztrációt követően a response tartalmazni fogja a User modelt és annak a tokenjét.  
Ha ez teljesült az Item-ek listázásához navigál az alkalmazás a navigateToGetItemsPage() metódus hívásával.

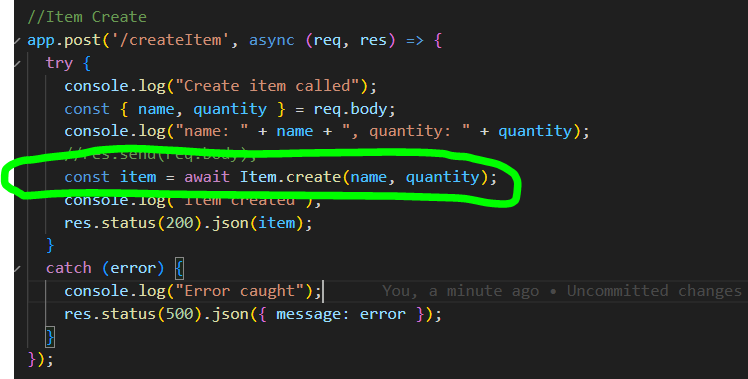


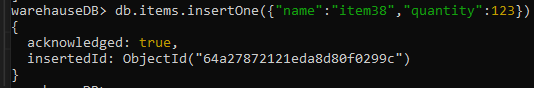


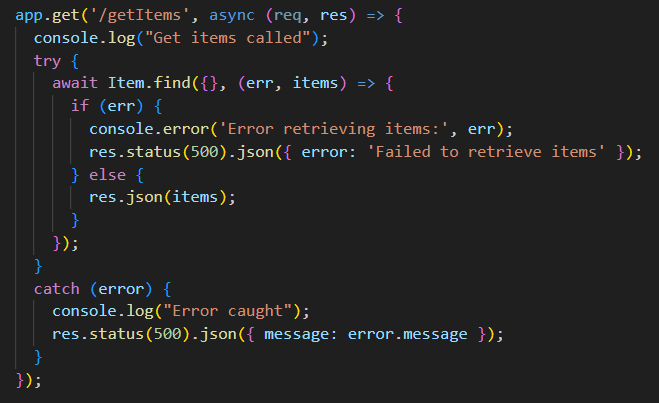


Miután sikerült működésre bírni az authentikációt létrehoztam a tárolt itemekhez tartozó CRUD elveknek megfelelő komponenseket (Item-Create, Item-List, Item-Update). A listázás tartalmazza a törlés funkciót is. Viszont amikor létrehoztam az Item modeljét és a mongoDB-be fel akartam vinni az Item.Create({„name”:”Item1,”quantity”:1})-et **teljesen megakadtam**.

Az await Item.create(name,quantity)-ből a legtöbb információ amit ki tudtam hozni, hogy   
"Cannot read properties of undefined (reading 'find')”.

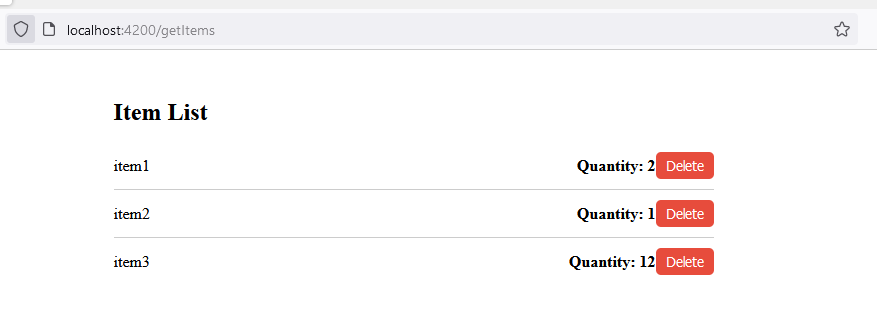
„Pár óra alatt” ahogy sikerült létrehozni a DB kapcsolatot és az authentikáció, bátran raktam félre az alkalmazást, hogy „szombat délután befejezem”, ugyanakkor a szombat kevésnek bizonyult.  
Vasárnap csak este tudtam vele foglalkozni és így 2 óra alvás követően teljesen tanácstalanul nézem a kódot, hogy mi felett siklottam vajon el?  
  
Valamiért mintha az Item modellemmel nem tudna mit kezdeni. Ugyanakkor mivel a User model mintájára hoztam létre, nem értem, hogy mit ronthattam el.  
  
  


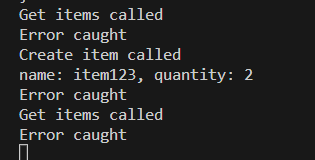
Természetesen következő lépésem volt, hogy MongoSH-t használva töltök fel itemeket.  


és a GetItems()-et megírva biztos jó lesz.  
  


Ugyanakkor szintén az Items.find({})-nál ugyan azt a hibát kaptam.

Elkezdtem ugyan a getItems megjelenésével foglalkozni, de nem hagyott nyugodni a dolog.





Ez a tömör kis Console.Log(„Error caught”) bizony beleégett a retinámba.  
  
Bocsánat, hogy csak most a kiírt vizsga időpontja előtt percekkel küldöm a beszámolót, de sosem lehet tudni, hogy mikor sikerül az áttörés. Bíztam benne, hogy hamar megtalálom.