Adatbázis: drinkisland

Táblák:

* Alapanyagok: Az alapanyagok nevét és mennyiségének mértékegységét tárolja.
* Italok: A koktélok nevét tárolja.
* Posts: A posztok írójának id-jét, a poszt címét és a szövegét tárolja.
* Receptek: Az italokhoz a recepteket tárolja.
* Tipus: Az italok típusát tárolja.
* Users: A felhasználók adatait tárolja.

alapanyagok:

* id integer
* name varchar(50)
* unit varchar(50)

italok:

* id integer
* name varchar(50)

posts:

* id integer
* user\_id integer
* title varchar(50)
* text varchar(50)

receptek:

* id integer
* ital\_id integer
* alapanyag\_id integer
* amount integer

tipus:

* id integer
* ital\_id integer
* teli tinyint
* nyari tinyint
* edes tinyint
* savanyu tinyint
* keseru tinyint
* alkoholos tinyint
* alkoholmentes tinyint

users:

* id integer
* username varchar(255)
* email varchar(255)
* password varchar(255)
* password\_confirmation varchar(255)
* remember\_token varchar(100)
* created\_at timestamp
* updated\_at timestamp
* login\_attempts integer
* banned\_time timestamp
* is\_admin tinyint

Kapcsolatok:

* A posts tábla user\_id mezője a users tábla id mezőjére hivatkozik.
* A receptek tábla ital\_id mezője az italok tábla id mezőjére hivatkozik.
* A receptek tábla alapanyag\_id mezője az alapanyagok tábla id mezőjére hivatkozik.
* A tipus tábla ital\_id mezője az italok tábla id mezőjére hivatkozik.

Kontrollerek:

* Auth/AuthController
* AlapanyagController
* ItalController
* KoktelController
* PostController
* ProfileController
* ReceptController
* TipusController

Eljárások:

AuthController:

register():

A ”/register” végponton érhető el, felhasználók regisztálását segítő eljárás. A felhasználó adatait a users tábába menti el. Paraméternek egy Request-et vár.

Request adatok:

* username varchar(255)
* email varchar(255)
* password varchar(255)
* password\_confirmation varchar(255)

Hogyha nem létezik ilyen felhasználónevű user, az e-mail cím megfelelő formátumú és a jelszók megegyeznek, akkor felveszi a felhasználót a users táblába.

login()

A ”/login” végponton érhető el, felhasználók bejelentkezését segítő eljárás. Paraméternek egy Request-et vár.

Request adatok:

* username varchar(255)
* password varchar(255)

Hogyha létezik ilyen felhasználónevű user, és a jelszó megfelelő, akkor belép a felhasználóval, és generál egy tokent a felhasználónak, amivel azonosíthatja magát, és visszaadja ezt.

logout()

A ”/ logout” végponton érhető el, hogyha a felhasználó be van jelentkezve. A felhasználók kijelentkezését segítő eljárás. Paraméternek egy Request-et vár, amiben a bejelentkezéskor kapott tokent várja.

Futtatásakor, ha a felhasználó megfelelő tokent ad vissza akkor kijelentkezik a felhasználó és törli ezt a tokent.

ProfileController:

getUserProfileData()

A /getUserProfileData végponton érhető el akkor, hogyha egy felhasználó be van jelentkezve. A felhasználó felhasználónevét adja vissza.

getUsers()

A /getUsers végponton érhető el, hogyha egy admin van bejelentkezve.

Ha meghívjuk, akkor visszatér az összes felhasználó összes adatával a users táblából.

setNewPassword()

A /setNewPassword végpontról érhető el, ha a felhasználó be van jelentkezve.

Request adatok:

* password varchar(255)
* password\_confirmation varchar(255)

Ha meghívjuk, és átadjuk a bejelentkezéskor kapott tokenünket, akkor megváltoztatja a felhasználó jelszavát.

setAdmin()

A /setAdmin végponton érhető el, ha adminként van a felhasználó bejelentkezve, és bemenetként egy Requestet vár.

Request adatok:

* username varchar(255)

Ha a felhasználó admin, és létezik username nevű felhasználó, akkor admin hozzáférést ad neki.

deleteAccount()

A /deleteAccount végponton érthető el, ha a felhasználó be van jelentkezve.

Ha meghívja a felhasználó, és megfelelő tokent ad át, akkor törli a felhasználót.

AlapanyagController:

addAlapanyag()

A /addAlapanyag végponton érhető el, ha admin van bejelentkezve, és egy Requestet vár.

Request adatok:

* name varchar(50)
* unit varchar(50)

Ha olyan alapanyagnevet kapunk, ami még nem létezik, akkor az alapanyagok táblához hozzáad egy újabb alapanyagot és a mértékegységét.

getAlapanyagByName()

A /getAlapanyagByName végponton érjük el, és egy Requestet vár.

Request adatok:

* name varchar(50)

Ha létező nevet adunk meg, akkor visszaadja az ahhoz az névhez tartozó alapanyagot.

getAlapanyagById()

A /getAlapanyagById végponton érjük el, és egy Requestet vár.

Request adatok:

* id integer

Ha létező id-t adunk meg, akkor visszaadja az ahhoz az id-hez tartozó alapanyagot.

getAlapanyagok()

A /getAlapanyagok végponton érjük el.

Visszaadja az összes alapanyagot.

modifyAlapanyag()

A /modifyAlapanyag végponton érjük el, és egy Requestet vár, ahol id és egy name attribútumokat várja, ahol a name az új név.

Request adatok:

* id integer
* name varchar(50)
* unit varchar(50)

Ha létező id-t adunk meg, és nincs még olyan név, amire meg szeretnénk változtatni, akkor megváltoztatja az alapanyagok táblában az adatokat.

deleteAlapanyag()

A /deleteAlapanyag végponton érjük el, ha adminként vagyunk bejelentkezve, és egy Requestet vár.

Request adatok:

* id integer

Ha létezik ilyen id-jű alapanyag, és egyik receptben sincs összetevőként bent, akkor törli az alapanyagot. Azt, hogy benne van-e egy receptben a Receptkontroller getReceptByAlapanyagId() metódussal lehet lekérdezni, ami egy id-t kér Request-ként.

ItalController

addItal()

A /addItal végponton lehet elérni, ha admin van bejelentkezve a felhasználó és egy Requestet vár.

Request adatok:

* name varchar(50)

Ha olyan italnevet adunk, ami még nem létezik, akkor az italok táblához hozzáad egy újabb ital.

getItalByName()

A /getItalByName végponton érjük el, és egy Requestet vár.

Request adatok:

* name varchar(50)

Ha létező nevet adunk meg, akkor visszaadja az ahhoz az névhez tartozó italt.

getItalById()

A /getItalById végponton érjük el, és egy Requestet vár.

Request adatok:

* id integer

Ha létező id-t adunk meg, akkor visszaadja az ahhoz az id-hez tartozó italt.

getItalok()

A /getItalok végponton érjük el.

Visszaadja az összes italt.

modifyItal ()

A /modifyItal végponton érjük el, és egy Requestet vár, ahol id és egy name attribútumokat várja, ahol a name az új név.

Request adatok:

* id integer
* name varchar(50)
* unit varchar(50)

Ha létező id-t adunk meg, és nincs még olyan név, amire meg szeretnénk változtatni, akkor megváltoztatja az italok táblában az adatokat.

deleteItal ()

A /deleteItal végponton érjük el, ha adminként vagyunk bejelentkezve, és egy Requestet vár.

Request adatok:

* id integer

Ha létezik ilyen id-jű ital, és egyik receptben sincs összetevőként bent, akkor törli az italt. A receptek táblából kitörli az összes olyan receptet, ami ehhez az id-hez tartozik, és a tipusok táblából pedig a hozzá tartozó típust A TipusController deleteTipusByItalId() metódusával.

KoktelController

getKoktelok()

A /getKoktelok végponton érhető el, és a koktélok receptjeit adja vissza.

Egy json file-okat tartalmazó tömböt ad vissza, a következő példán látható módon:

[

{

"Ital": "Limonádé",

"Recept": {

"citromlé": "2 cl",

"szódavíz": "20 cl",

"Jégkocka": "6 db",

"mentalevél": "1 db",

"lime": "2 szelet",

"cukorszirup": "2 cl"

}

},

{

"Ital": "APEROL SPRITZ",

"Recept": {

"narancskarika": "1 szelet",

"szóda": "4 cl",

"prosecco": "12 cl",

"Jégkocka": "1 db"

}

}

]

searchKoktelokByName()

A /searchKoktelokByName végponton érjük el, és egy Requestet vár.

Request adatok:

* text varchar

Az előzőhöz hasonló json fájlokat tartalmazó tömböt ad vissza, csak azokat az italokat tartalmazva, amiknek a nevében benne van a Requestben kapott text paraméter.

getReceptByName()

A /getReceptByName végponton lehet elérni, és egy Requestre vár.

Request adatok:

* name varchar(50)

Egy JSON fájlt ad vissza, amiben annak a koktélnak a receptje van, amelyiknek a nevét a name attribútumban kapta.

getTeli()