Használt elemek

Készítették: Madai Bálint, Szabó-Holman Dániel 13.c

# Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc183433493)

[Frontend és Backend 4](#_Toc183433494)

[Osztályok 4](#_Toc183433495)

[Frontend 6](#_Toc183433496)

[Függvények 6](#_Toc183433497)

[Általános 6](#_Toc183433498)

[Regisztráció 7](#_Toc183433499)

[Bejelentkezés 7](#_Toc183433500)

[Elfelejtett jelszó 7](#_Toc183433501)

[Recept feltöltése 7](#_Toc183433502)

[Recept interaktív funkciók 8](#_Toc183433503)

[Kedvencek 8](#_Toc183433504)

[Kedvelések 8](#_Toc183433505)

[Kommentek 9](#_Toc183433506)

[Admin felület és saját profil 9](#_Toc183433507)

[Admin felület és saját receptek 10](#_Toc183433508)

[Eljárások 10](#_Toc183433509)

[Általános 10](#_Toc183433510)

[Backend 12](#_Toc183433511)

[Függvények 12](#_Toc183433512)

[Általános 12](#_Toc183433513)

[Regisztráció 13](#_Toc183433514)

[Bejelentkezés 13](#_Toc183433515)

[Recept feltöltése 13](#_Toc183433516)

[Recept interaktív funkciók 13](#_Toc183433517)

[Kedvencek 13](#_Toc183433518)

[Kedvelések 14](#_Toc183433519)

[Kommentek 15](#_Toc183433520)

[Admin felület és saját profil 15](#_Toc183433521)

[Admin felület és saját receptek 16](#_Toc183433522)

[Elfelejtett jelszó 16](#_Toc183433523)

[Eljárások 16](#_Toc183433524)

[Elfelejtett jelszó 16](#_Toc183433525)

# Frontend és Backend

## Osztályok

Role

* id[int]
* name[string]

User

* id[int?]
* name[string]
* email[string]
* password[string?]
* roleId[int?]

Commodity

* id[int]
* name[string]

Measure

* id[int]
* name[string]

Ingredient

* recipeId[int]
* commodityId[int]
* measureId[int]
* quantity[int]

Recipe

* recipeName[string?]
* images[Array<image>?]
* ingredients[Array<Ingredient>?]
* preparationTime[Date?]
* preparationDescription[string?]
* uploadDateTime[Date?]
* likeCount[int?]
* stateId[int?]

Favorite

* userId[int]
* recipeId[int]

Like

* userId[int]
* recipeId[int]

Comment

* id[int]
* userId[int]
* recipeId[int]
* text[string]

# Frontend

## Függvények

### Általános

GetUserById

* **Bemenet:** ***userId***[int]
* **Kimenet: *user***[User]
* **Működés:** Meghívja a backend enpointot, és átadja neki a ***userId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld.

GetRecipeById

* **Bemenet: *recipeId***[int]
* **Kimenet:** ***recipe***[Recipe]
* **Működés:** Meghívja a backend enpointot, és átadja neki a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld.

GetRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Meghívja a backend enpointot. Visszatérési értéke az lesz, amit az endpoint visszaküld.

SearchRecipes

* **Bemenet: *query***[string]
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Meghívja a backend enpointot, és átadja neki a ***query***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld.

GetPopularRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Meghívja a backend enpointot. Visszatérési értéke az lesz, amit az endpoint visszaküld.

GetFreshRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Meghívja a backend enpointot. Visszatérési értéke az lesz, amit az endpoint visszaküld.

### Regisztráció

RegisterUser

* **Bemenet:** ***user***[User]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott felhasználó, akkor meghívja a backend endpointot, és átadja neki a ***user*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

PasswordMatches

* **Bemenet:** ***password***[string], ***confirmPassword***[string]
* **Kimenet:** ***matches***[boolean]
* **Működés:** Megnézi, hogy egyezik-e a két paraméterként megadott jelszó. Ha igen, ***true***-t, ha nem, ***false***-t ad vissza.

### Bejelentkezés

LogInUser

* **Bemenet:** ***user***[User]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és annak válasza alapján bejelentkezteti a felhasználót, ha létezik. Visszatérési értéke ***true*** lesz, ha sikerült bejelentkeztetni a felhasználót, ***false***, ha nem.

### Elfelejtett jelszó

UserForgotPassword

* **Bemenet:** ***email***[string]
* **Kimenet: *exists***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadj neki az ***email***t. Visszatérési értéke az lesz, amit az endpoint visszaküld (létezik felhasználó: ***true***, nem létezik felhasználó: ***false***).

### Recept feltöltése

UploadRecipe

* **Bemenet:** ***recipe***[Recipe]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott recept, akkor meghívja a backend endpointot, és átadja neki a ***recipe*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

### Recept interaktív funkciók

#### Kedvencek

GetFavoritesByUserId

* **Bemenet: *userId***[int]
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t. A ***recipes***-be beletölti a válaszként kapott tömböt, és visszaadja azt.

AlreadyFavorite

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***already***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t, amire válaszul megkapja, hogy van-e már rekord az adatbázis ***favorites*** táblájában ezekkel az adatokkal. Ha van már, ***true***-t, ha nincs ***false***-t ad vissza.

FavoriteRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

UnFavoriteRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

#### Kedvelések

AlreadyLiked

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***already***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t, amire válaszul megkapja, hogy van-e már rekord az adatbázis ***likes*** táblájában ezekkel az adatokkal. Ha van már, ***true***-t, ha nincs ***false***-t ad vissza.

LikeRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

UnLikeRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***userId***-t és a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

#### Kommentek

GetCommentsByRecipeId

* **Bemenet: *recipeId***[int]
* **Kimenet:** ***comments***[Array<Comment>]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***recipeId***-t. A ***comments***-be beletölti a válaszként kapott tömböt, és visszaadja azt.

NewComment

* **Bemenet: *comment***[Comment]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***comment*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

EditComment

* **Bemenet: *comment***[Comment]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***comment*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

DeleteComment

* **Bemenet: *commentId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Meghívja a backend endpointot, és átadja neki a ***comment***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

### Admin felület és saját profil

EditUser

* **Bemenet: *user***[User]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha a bejelentkezett felhasználónak van joga ehhez a művelethez, és minden kritériumnak megfelelnek az adatok, akkor meghívja a backend endpointot, és átadja neki a ***user*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

DeleteUser

* **Bemenet:** ***userId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha a bejelentkezett felhasználónak van joga ehhez a művelethez, akkor meghívja a backend endpointot, és átadja neki a ***userId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

### Admin felület és saját receptek

EditRecipe

* **Bemenet:** ***recipe***[Recipe]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha a bejelentkezett felhasználónak van joga ehhez a művelethez, és minden kritériumnak megfelelnek az adatok, akkor meghívja a backend endpointot, és átadja neki a ***recipe*** objektumot. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

DeleteRecipe

* **Bemenet:** ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha a bejelentkezett felhasználónak van joga ehhez a művelethez, akkor meghívja a backend endpointot, és átadja neki a ***recipeId***-t. Visszatérési értéke az lesz, amit az endpoint visszaküld (sikerült: ***true***, nem sikerült: ***false***).

## Eljárások

### Általános

RenderRecipes

* **Bemenet: *-***
* **Kimenet:** ***-***
* **Működés:** Egy ciklusban tovább adja a ***recipes*** Array elemeit egyenként (mely a függvényt tartalmazó osztályban található), oldalanként legfeljebb 15 receptet, a RenderRecipe eljárásnak.

RenderRecipe

* **Bemenet:** ***recipe***[Recipe]
* **Kimenet:** ***-***
* **Működés:** Elkészíti a kapott ***recipe*** objektum alapján a recept kártyáját.

SwitchRenderedRecipes

* **Bemenet: *-***
* **Kimenet: *-***
* **Működés:** Vált a megjelenített receptek között.

# Backend

## Függvények

### Általános

GetUserById

* **Bemenet:** ***userId***[int]
* **Kimenet: *user***[User]
* **Működés:** Lekérdezi az adatbázis ***users*** táblájából, hogy létezik-e recept a paraméterként megadott ***userId*** azonosítóval. Ha létezik, annak adatait tölti bele a ***user*** objektumba, ha nem, üresen hagyja azt, és ezt az objektumot adja vissza.

GetRecipeById

* **Bemenet: *recipeId***[int]
* **Kimenet:** ***recipe***[Recipe]
* **Működés:** Lekérdezi az adatbázis ***recipes*** táblájából, hogy létezik-e recept a paraméterként megadott ***recipeId*** azonosítóval. Ha létezik, annak adatait tölti bele a ***recipe*** objektumba, ha nem, üresen hagyja azt, és ezt az objektumot adja vissza.

GetRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Lekérdezi az adatbázis ***recipes*** táblájából az összes receptet. Ezen receptek adatait tölti bele a ***recipes*** Array-ba, recipe objektumokként.

SearchRecipes

* **Bemenet: *query***[string]
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Lekérdezi az adatbázis ***recipes*** táblájából azokat a recepteket, melynek neve, vagy elkészítési leírása tartalmazza paraméterként kapott ***query***-t. Ezen receptek adatait tölti bele a ***recipes*** Array-ba, recipe objektumokként.

GetPopularRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array <Recipe>]
* **Működés:** Lekérdezi az adatbázis ***recipes*** táblájából a 9 legtöbb kedveléssel rendelkező receptet. Ezen receptek adatait tölti bele a ***recipes*** Array-ba, recipe objektumokként.

GetFreshRecipes

* **Bemenet: *-***
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Lekérdezi az adatbázis ***recipes*** táblájából a 9 legújabban feltöltött receptet. Ezen receptek adatait tölti bele a ***recipes*** Array-ba, recipe objektumokként.

### Regisztráció

RegisterUser

* **Bemenet:** ***user***[User]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott felhasználó, akkor beszúrja azt az adatbázis ***users*** táblájába. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

### Bejelentkezés

LogInUser

* **Bemenet:** ***user***[User]
* **Kimenet: *role***[Role]
* **Működés:** Lekérdezi az adatbázis ***users*** táblájából, hogy létezik-e felhasználó a paraméterként megadott adatokkal (az email címet és a jelszót veszi figyelembe). Ha létezik, annak hozzáférési szintjének adatait tölti bele a ***role*** objektumba, ha nem, üresen hagyja azt, és ezt az objektumot adja vissza.

### Recept feltöltése

UploadRecipe

* **Bemenet:** ***recipe***[Recipe]
* **Kimenet: *success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott receptet, akkor beszúrja azt az adatbázisba. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

### Recept interaktív funkciók

#### Kedvencek

GetFavoritesByUserId

* **Bemenet: *userId***[int]
* **Kimenet:** ***recipes***[Array<Recipe>]
* **Működés:** Lekéri az adatbázis ***favorites*** táblájából a ***userId*** alapján az adott felhasználó kedvelt receptjeit, és visszaadja azokat.

AlreadyFavorite

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***already***[boolean]
* **Működés:** Lekéri az adatbázis ***favorites*** táblájából, hogy létezik-e olyan rekord, amely a ***userId***-t és a ***recipeId***-t tartalmazza. Ha igen, ***true***-t, ha nem, ***false***-t ad vissza.

FavoriteRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***favorites*** táblájába beszúr egy új sort, mely a ***userId***-t és a ***recpieId***-t tartalmazza. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

UnFavoriteRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***favorites*** táblájából kitörli azt a sort, amely, amely a ***userId***-t és a ***recpieId***-t tartalmazza. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

#### Kedvelések

AlreadyLiked

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***already***[boolean]
* **Működés:** Lekéri az adatbázis ***likes*** táblájából, hogy létezik-e olyan rekord, amely a ***userId***-t és a ***recipeId***-t tartalmazza. Ha igen, ***true***-t, ha nem, ***false***-t ad vissza.

LikeRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***likes*** táblájába beszúr egy új sort, mely a ***userId***-t és a ***recpieId***-t tartalmazza. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

UnLikeRecipe

* **Bemenet: *userId***[int], ***recipeId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***likes*** táblájából kitörli azt a sort, amely a ***userId***-t és a ***recpieId***-t tartalmazza. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

#### Kommentek

GetCommentsByRecipeId

* **Bemenet: *recipeId***[int]
* **Kimenet:** ***comments***[Array<Comment>]
* **Működés:** Lekéri az adatbázis ***comments*** táblájából a ***recipeId*** alapján az adott recepthez tartozó kommenteket, és visszaadja azokat.

NewComment

* **Bemenet: *comment***[Comment]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***comments*** táblájába beszúr egy új sort a ***comment*** objektum adataival. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

EditComment

* **Bemenet: *comment***[Comment]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***comments*** táblájában módosítja az egyik sort a ***comment*** objektum alapján. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

DeleteComment

* **Bemenet: *commentId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Az adatbázis ***comments*** táblájából kitöröl egy sort a ***comment*** objektum alapján. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

### Admin felület és saját profil

EditUser

* **Bemenet:** ***user***[User]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott felhasználó, akkor módosítja az adatbázis ***users*** táblájában a paraméterként megadott felhasználó adatait az azonosítója alapján. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

DeleteUser

* **Bemenet: *userId***[int]
* **Kimenet:** ***success***[boolean]
* **Működés:** Ha létezik felhasználó a paraméterként megadott azonosítóval, akkor törli azt az adatbázis ***users*** táblájából. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

### Admin felület és saját receptek

EditRecipe

* **Bemenet:** ***recipe***[Recipe]
* **Kimenet: *success***[boolean]
* **Működés:** Ha minden kritériumnak megfelel a paraméterként megadott recept, akkor módosítja az adatbázis ***recipes*** táblájában a paraméterként megadott recept adatait az azonosítója alapján. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

DeleteRecipe

* **Bemenet:** ***recipeId***[int]
* **Kimenet: *success***[boolean]
* **Működés:** Ha létezik recept a paraméterként megadott azonosítóval, akkor törli azt az adatbázis ***recpies*** táblájából. A művelet sikerességétől függ a visszatérési érték (sikerült: ***true***, nem sikerült: ***false***).

### Elfelejtett jelszó

UserExistsWithEmail

* **Bemenet: *email***[string]
* **Kimenet: *exists***[boolean]
* **Működés:** Lekérdezi az adatbázis ***users*** táblájából, hogy létezik-e felhasználó a paraméterként megadott emaillel. Ha igen, akkor ***true***-t, ha nem, akkor ***false***-t ad vissza.

## Eljárások

### Elfelejtett jelszó

SendResetPasswordEmail

* **Bemenet:** ***email***[string]
* **Kimenet: *-***
* **Működés:** Meghívja a ***UserExistsWithEmail*** függvényt, paraméternek megadja az ***email***t, és annak visszatérési értékétől függően vagy küld egy emailt a jelszó visszaállításához, vagy nem tesz semmit.