

# RecipeBookApp Dokumentáció

## API Leírás

### Alap URL: /api

Az összes végpont előtt ez az alap elérési út szerepel.

### Authentikáció

A rendszer JWT (JSON Web Token) alapú autentikációt használ. Bizonyos végpontok csak autentikált felhasználók számára érhetőek el, míg mások publikusak.

### 1. Authentikációs végpontok

#### 1.1 Bejelentkezés

- **Végpont:** POST /auth/login
- **Leírás:** Felhasználó bejelentkezése
- **Bemeneti paraméter:**

```
{  "username": "string",  "password": "string"}
```
- **Válasz:** JWT token (string)
- **Hozzáférés:** Publikus

#### 1.2 Regisztráció

- **Végpont:** POST /auth/register
- **Leírás:** Új felhasználó regisztrálása
- **Bemeneti paraméter:**

```
{  "username": "string",  "password": "string"}
```
- **Válasz:** "User registered successfully" (string)
- **Hozzáférés:** Publikus

#### 1.3 Token ellenőrzés

- **Végpont:** GET /auth/check
- **Leírás:** JWT token érvényességének ellenőrzése
- **Válasz:** 200 OK (ha a token érvényes)
- **Hozzáférés:** Authentikált felhasználók

## 1.4 Admin jogosultság ellenőrzés

- **Végpont:** GET /auth/checkAdminRole
- **Leírás:** Admin jogosultság ellenőrzése
- **Válasz:** 200 OK (ha a felhasználó admin)
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 2. Recept kezelési végpontok

### 2.1 Receptek keresése/listázása

- **Végpont:** GET /recipe
- **Leírás:** Receptek keresése és lapozása
- **Bemeneti paraméter:** (query paraméterek) és Pageable (lapozási paraméterek) RecipeFilter
- **Válasz:** Page<RecipeDto> - lapozott recept lista
- **Hozzáférés:** Publikus

### 2.2 Recept részleteinek lekérése

- **Végpont:** GET /recipe/{id}
- **Leírás:** Egy adott recept részletes adatainak lekérése azonosító alapján
- **Bemeneti paraméter:** id (Long) - a recept azonosítója
- **Válasz:** - a recept részletes adatai RecipeDto
- **Hozzáférés:** Publikus

### 2.3 Új recept hozzáadása

- **Végpont:** POST /recipe
- **Leírás:** Új recept létrehozása a rendszerben
- **Bemeneti paraméter:** - a recept adatai RecipeDto
- **Válasz:** - a létrehozott recept adatai RecipeDto
- **Hozzáférés:** Authentikált felhasználók

### 2.4 Recept módosítása

- **Végpont:** PUT /recipe/{id}
- **Leírás:** Meglévő recept adatainak módosítása
- **Bemeneti paraméterek:**
  - id (Long) - a recept azonosítója
  - - a recept módosított adatai RecipeDto
- **Válasz:** - a módosított recept adatai RecipeDto
- **Hozzáférés:** Authentikált felhasználók (saját recept esetén)

## 2.5 Recept törlése

- **Végpont:** DELETE /recipe/{id}
- **Leírás:** Recept törlése azonosító alapján
- **Bemeneti paraméter:** id (Long) - a recept azonosítója
- **Válasz:** Nincs (204 No Content)
- **Hozzáférés:** Authentikált felhasználók (saját recept esetén)

## 2.6 Saját receptek lekérése

- **Végpont:** GET /recipe/myrecipes
- **Leírás:** A bejelentkezett felhasználó saját receptjeinek lekérése
- **Válasz:** List<RecipeDto> - a felhasználó receptjeinek listája
- **Hozzáférés:** Authentikált felhasználók

## 2.7 Recept értékelés lekérése

- **Végpont:** GET /recipe/{id}/rating
- **Leírás:** Egy adott recept átlagos értékelésének lekérése
- **Bemeneti paraméter:** id (Long) - a recept azonosítója
- **Válasz:** Átlagos értékelés (double)
- **Hozzáférés:** Publikus

## 2.8 Recept értékelés hozzáadása/módosítása

- **Végpont:** PUT /recipe/{id}/rating
- **Leírás:** Recept értékelése vagy meglévő értékelés módosítása
- **Bemeneti paraméterek:**
  - id (Long) - a recept azonosítója
  - value (Integer) - értékelési érték
- **Válasz:** - az értékelés adatai RatingDto
- **Hozzáférés:** Authentikált felhasználók

## 2.9 Szűrők lekérése

- **Végpont:** GET /recipe/filters
- **Leírás:** A receptek szűrésére használható szűrők lekérése
- **Válasz:** - a lehetséges szűrési paraméterek RecipeFilter
- **Hozzáférés:** Publikus

## 2.10 Hozzávalók lekérése

- **Végpont:** GET /recipe/ingredients

- **Leírás:** Az összes rendszerben szereplő hozzávaló lekérése
- **Válasz:** Set<String> - egyedi hozzávalók listája
- **Hozzáférés:** Publikus

### 3. Felhasználói végpontok

#### 3.1 Felhasználói profil lekérése

- **Végpont:** GET /user/profile
- **Leírás:** A bejelentkezett felhasználó profiljának lekérése
- **Válasz:** Felhasználónév (string)
- **Hozzáférés:** Authentikált felhasználók

#### 3.2 Felhasználónév módosítása

- **Végpont:** PUT /user/update/username
- **Leírás:** A bejelentkezett felhasználó felhasználónevének módosítása
- **Bemeneti paraméter:**

```
{
  "username": "string"
}
```
- **Válasz:** Új JWT token (string)
- **Hozzáférés:** Authentikált felhasználók

#### 3.3 Jelszó módosítása

- **Végpont:** PUT /user/update/password
- **Leírás:** A bejelentkezett felhasználó jelszavának módosítása
- **Bemeneti paraméter:**

```
{
  "currentPassword": "string",
  "newPassword": "string"
}
```
- **Válasz:** Új JWT token (string)
- **Hozzáférés:** Authentikált felhasználók

### 4. Admin végpontok

#### 4.1 Felhasználók listázása

- **Végpont:** GET /admin/users
- **Leírás:** Összes felhasználó listázása
- **Válasz:** List<UserDto> - felhasználók listája
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 4.2 Felhasználó törlése

- **Végpont:** DELETE /admin/users/{userId}
- **Leírás:** Felhasználó törlése azonosító alapján
- **Bemeneti paraméter:** userId (Long) - felhasználó azonosítója
- **Válasz:** Nincs (204 No Content)
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 4.3 Felhasználói szerepkör módosítása

- **Végpont:** PUT /admin/users/{userId}/role
- **Leírás:** Felhasználó szerepkörének módosítása
- **Bemeneti paraméterek:**
  - userId (Long) - felhasználó azonosítója
  - role (String) - új szerepkör
- **Válasz:** Nincs (204 No Content)
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 4.4 Összes recept listázása

- **Végpont:** GET /admin/recipes
- **Leírás:** Az összes recept egyszerűsített adatainak listázása
- **Válasz:** List<RecipeInfoLineDto> - receptek egyszerűsített adatai
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 4.5 Recept törlése (admin)

- **Végpont:** DELETE /admin/recipes/{recipeId}
- **Leírás:** Bármely recept törlése admin jogkörrel
- **Bemeneti paraméter:** (Long) - recept azonosítója recipeId
- **Válasz:** Nincs (204 No Content)
- **Hozzáférés:** Admin szerepkörrel rendelkező felhasználók

## 5. Komment kezelési végpontok

### 5.1 Recept kommentek lekérése

- **Végpont:** GET /comment/{recipeId}
- **Leírás:** Egy adott recepthez tartozó kommentek lekérése
- **Bemeneti paraméter:** (Long) - a recept azonosítója recipeId
- **Válasz:** List<CommentDto> - kommentek listája
- **Hozzáférés:** Publikus

## 5.2 Komment hozzáadása recepthez

- **Végpont:** POST /comment/{recipeId}
- **Leírás:** Új komment hozzáadása egy recepthez
- **Bemeneti paraméterek:**
  - (Long) - a recept azonosítója recipeId
  - - a komment adatai CommentDto
- **Válasz:** - a létrehozott komment adatai CommentDto
- **Hozzáférés:** Authentikált felhasználók

## 6. Adattípusok

### UserDto

```
{
  "id": 1,
  "username": "felhasznalonev",
  "password": "jelszo",
  "roles": ["USER", "ADMIN"]
}
```

### RecipeDto

```
{
  "id": 1,
  "title": "Recept címe",
  "ingredients": [
    {
      "name": "hozzávaló neve",
      "amount": "mennyiség"
    }
  ],
  "instructions": "Az elkészítés lépései...",
  "createdAt": "2023-05-15",
  "image": "base64 kódolt kép",
  "cookingTime": 30,
  "author": "szerző neve"
}
```

### RecipeInfoLineDto

```
{
  "id": 1,
  "title": "Recept címe",
  "createdAt": "2023-05-15",
  "author": "szerző neve"
}
```

### IngredientDto

```
{
  "name": "hozzávaló neve",
}
```

```
"amount": "mennyiség"
}
```

#### **CommentDto**

```
{
  "id": 1,
  "text": "Komment szövege",
  "user": "felhasználó neve",
  "createdAt": "2023-05-15"
}
```

#### **RatingDto**

```
{
  "id": 1,
  "user": "felhasználó neve",
  "recipeId": 123,
  "value": 5
}
```

#### **UsernameUpdateDTO**

```
{
  "username": "új felhasználónév"
}
```

#### **PasswordUpdateDTO**

```
{
  "currentPassword": "jelenlegi jelszó",
  "newPassword": "új jelszó"
}
```

#### **RecipeFilter**

```
{
  "createdFrom": "2023-01-01",
  "createdTo": "2023-12-31",
  "ingredientNames": ["liszt", "cukor", "tojás"],
  "cookingTimeFrom": 10,
  "cookingTimeTo": 60,
  "minRating": 4.5,
  "searchTerm": "keresési kifejezés"
}
```

### **7. Biztonsági megfontolások**

Az API a következő biztonsági beállításokkal rendelkezik:

- JWT alapú autentikáció
- Bizonyos végpontok csak autentikált felhasználók számára elérhetők
- Admin végpontok csak admin jogosultsággal rendelkező felhasználók számára elérhetők
- Jelszavak BCrypt titkosítással tárolódnak

- CORS beállítások a frontend számára

## Entitások

Az alkalmazás adatmodellje 5 fő entitásból áll, amelyek a receptek kezelését, a felhasználói interakciókat és az értékeléseket biztosítják.

### 1. User (Felhasználó)

Az entitás az alkalmazás felhasználóit reprezentálja.

#### Tulajdonságok:

- id (Long): Egyedi azonosító
- username (String): Felhasználónév
- password (String): Jelszó (titkosítva tárolva)
- roles (Set): Szerepkörök (USER, ADMIN)

#### Kapcsolatok:

- recipes (Set): A felhasználó által létrehozott receptek (@OneToMany)
- comments (Set): A felhasználó által írt hozzászólások (@OneToMany)
- ratings (Set): A felhasználó által adott értékelések (@OneToMany)

### 2. Recipe (Recept)

Az entitás a recepteket reprezentálja.

#### Tulajdonságok:

- id (Long): Egyedi azonosító
- title (String): A recept címe
- instructions (String): Elkészítési útmutató
- createdAt (LocalDate): Létrehozás dátuma
- image (byte []): A recepthez tartozó kép
- cookingTime (int): Elkészítési idő percben

#### Kapcsolatok:

- author (User): A recept szerzője (@ManyToOne)
- ingredients (Set): A recepthez tartozó hozzávalók (@OneToMany)
- comments (List): A recepthez írt hozzászólások (@OneToMany)
- ratings (List): A recepthez adott értékelések (@OneToMany)

### 3. Ingredient (Hozzávaló)

Az entitás a receptek hozzávalóit reprezentálja.

#### Tulajdonságok:



- id (Long): Egyedi azonosító
- name (String): Hozzávaló neve
- amount (String): Mennyiség (pl. "2 ek", "100 g")

**Kapcsolat:**

- recipe (Recipe): A recept, amihez a hozzávaló tartozik (@ManyToOne)

**4. Comment (Hozzászólás)**

Az entitás a receptekhez fűzött hozzászólásokat reprezentálja.

**Tulajdonságok:**

- id (Long): Egyedi azonosító
- text (String): A hozzászólás szövege
- createdAt (LocalDate): Létrehozás dátuma

**Kapcsolatok:**

- user (User): A hozzászólás szerzője (@ManyToOne)
- recipe (Recipe): A recept, amihez a hozzászólás tartozik (@ManyToOne)

**5. Rating (Értékelés)**

Az entitás a receptekre adott értékeléseket reprezentálja.

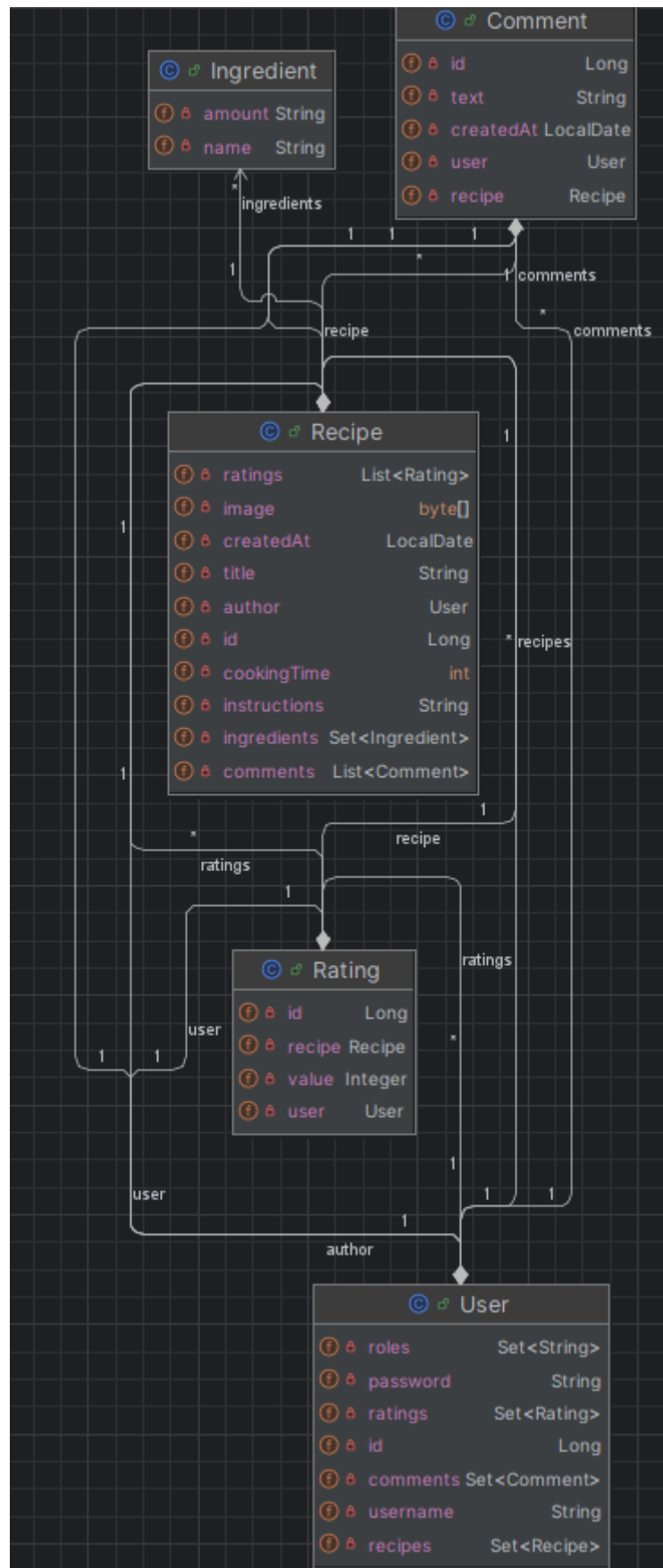
**Tulajdonságok:**

- id (Long): Egyedi azonosító
- value (Integer): Az értékelés értéke (1-5)

**Kapcsolatok:**

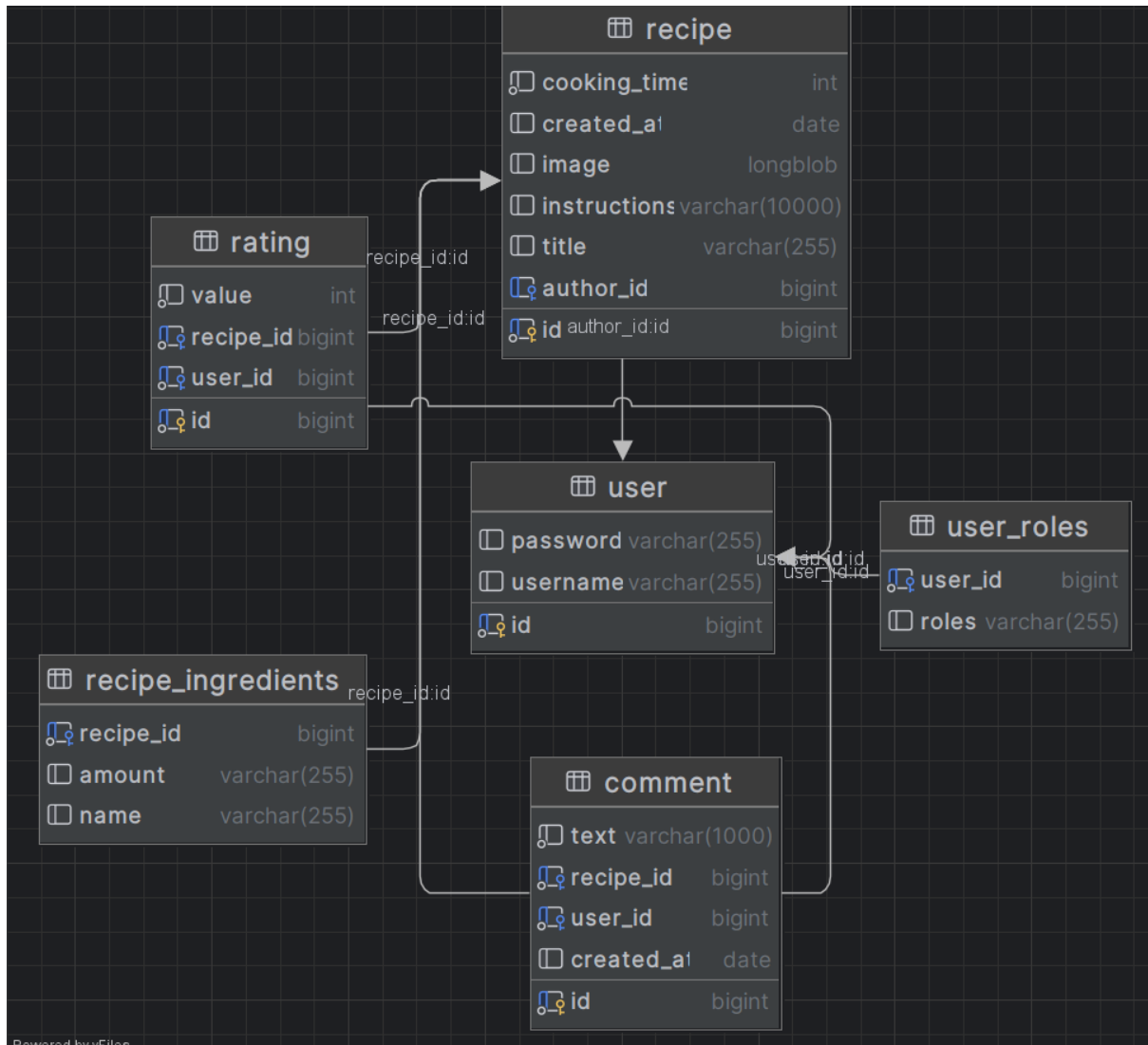
- user (User): Az értékelést adó felhasználó (@ManyToOne)
- recipe (Recipe): Az értékelt recept (@ManyToOne)

## Osztálydiagram:



## ORM/Adatbázis táblák

A Spring Data JPA és a Hibernate segítségével az entitások automatikusan leképződnek adatbázis táblákra. Az alábbi diagramon látható, hogyan alakulnak ki az adatbázis táblák az entitások alapján.



## UI navigáció

Az alkalmazás a React Router könyvtárat használja a navigációhoz és az URL alapú útvonalkezeléshez. Ez lehetővé teszi a Single Page Application (SPA) struktúra kialakítását, ahol az oldal újratöltése nélkül navigálhatunk különböző nézetek között.

## **Navigáció alapjai**

Az alkalmazás a következő kulcsfontosságú React Router elemeket használja:

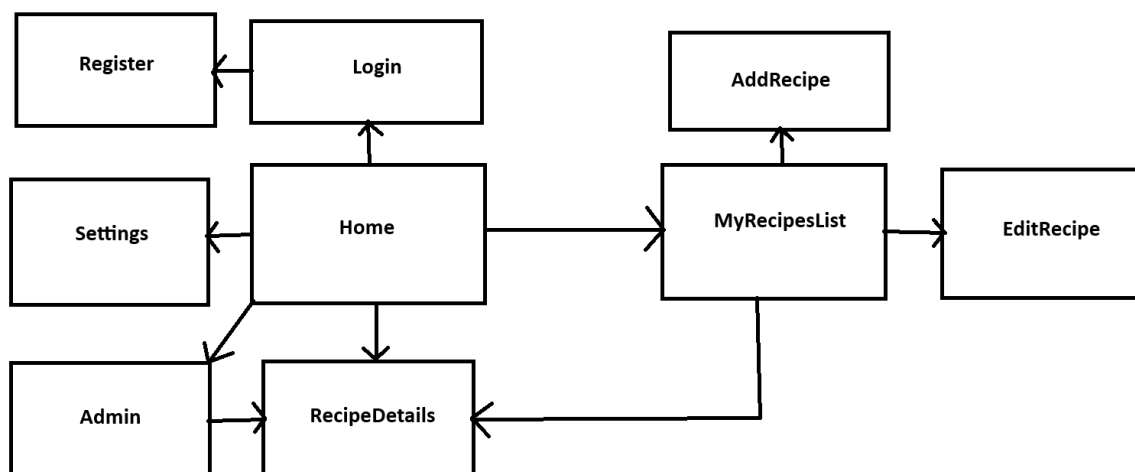
1. **BrowserRouter:** Az alkalmazás gyöker komponensét csomagolja, biztosítva az URL-alapú navigációt.
2. **Routes és Route:** Definiálják az alkalmazás különböző útvonalait és az azokhoz tartozó komponenseket.

### 3. **useNavigate hook:** Programozott navigációt tesz lehetővé komponenseken belül.

Az alkalmazásban az alábbi útvonalak találhatók:

Útvonal	Komponens	Funkció
/login	<Login />	Bejelentkezési oldal
/register	<Register />	Regisztrációs oldal
/my-recipes	<MyRecipesList />	A felhasználó saját receptjeinek listája
/add-recipe	<AddRecipe />	Új recept hozzáadása
/edit-recipe/:id	<EditRecipe />	Meglévő recept szerkesztése (paraméterezett ID-val)
/recipe/:id	<RecipeDetails />	Egy recept részletes nézete (paraméterezett ID-val)
/admin	<Admin />	Admin felület
/settings	<Settings />	Felhasználói beállítások
/	<Home />	Kezdőlap, alapértelmezett útvonal

A komponensek közti navigációt az alábbi diagram mutatja be:



A nyilak iránya az alkalmazásban a navigálás jellemző irányát mutatják, de vissza lépésre is van lehetőség az ellentétes irányban.

## UI Service-k

### 1. **recipeservice.js**

#### Áttekintés

A `recipeservice.js` modul a receptekkel kapcsolatos összes API kommunikációt kezeli. Lehetőséget biztosít receptek lekérdezésére, létrehozására, módosítására, értékelésére és törlésére.

#### API végpontok

- Alap URL: `http://localhost:8080/api/recipe`

Függvény	Leírás	Paraméterek	Visszatérési érték
getRecipes	Receptek lekérdezése szűrési, lapozási és rendezési opciókkal	filter: objektum, page: szám, size: szám, sort: szöveg	{ data: recepttömb, metadata: { totalPages, totalElements, size, number } }
getUserRecipes	Bejelentkezett felhasználó receptjeinek lekérdezése	filter: objektum	Receptek tömbje
getDefaultRecipeFilters	Alapértelmezett szűrési opciók lekérdezése	-	Szűrési opciók objektum
getIngredients	Elérhető hozzávalók lekérdezése	-	Hozzávalók tömbje
addRecipe	Új recept létrehozása	recipe: recept objektum	Létrehozott recept objektum
getRecipeById	Recept lekérdezése azonosító alapján	id: szám	Recept objektum
deleteRecipeById	Recept törlése azonosító alapján	id: szám	Törlés visszaigazolás
updateRecipe	Recept frissítése	id: szám, recipe: recept objektum	Frissített recept objektum
getRecipeRating	Recept értékelésének lekérdezése	id: szám	Értékelési adatok
submitRating	Recept értékelése	id: szám, ratingValue: szám	Frissített értékelési adatok

### Autentikáció

- A legtöbb végpont JWT token autentikációt igényel, amit a localStorage-ból olvas ki
- Az Authorization: Bearer \${token} fejléct a védett végpontokhoz minden kérésnél hozzáadja

## 2. authservice.js

### Áttekintés

Az authservice.js modul a felhasználói autentikációt és profilkezelést szolgálja, beleértve a bejelentkezést, kijelentkezést, és a felhasználói adatok módosítását.

### API végpontok

- Auth API: <http://localhost:8080/api/auth/>
- User API: <http://localhost:8080/api/user/>

Függvény	Leírás	Paraméterek	Visszatérési érték
login	Bejelentkezés	username: szöveg, password: szöveg	JWT token
logout	Kijelentkezés	-	-
saveToken	Token mentése	token: szöveg	-
getToken	Token lekérdezése	-	JWT token szöveg

checkAuth	Felhasználói autentikáció ellenőrzése	-	true/false
checkAdminRole	Admin jogosultság ellenőrzése	-	true/false
getUserProfile	Felhasználói profil adatok lekérése	-	Felhasználói profil objektum
updateUsername	Felhasználónév módosítása	newUsername: szöveg	Frissített JWT token
updatePassword	Jelszó módosítása	currentPassword: szöveg, newPassword: szöveg	Frissített JWT token

#### Biztonság

- A token localStorage-ban tárolódik
- Az autentikációs állapot ellenőrzése távoli API híváson keresztül történik
- A védett funkciók ellenőrzik a token meglétét műveletek előtt

### 3. adminservice.js

#### Áttekintés

Az adminservice.js modul adminisztrátori funkciókat biztosít a felhasználók és receptek kezelésére, amelyek csak admin jogosultsággal rendelkező felhasználók számára érhetők el.

#### API végpont

- Admin API: <http://localhost:8080/api/admin>

Függvény	Leírás	Paraméterek	Visszatérési érték
getAllUsers	Minden felhasználó lekérdezése	-	Felhasználók tömbje
deleteUser	Felhasználó törlése	userId: szám	Törlés visszaigazolás
updateUserRole	Felhasználói szerepkör módosítása	userId: szám, role: szöveg	Módosítás visszaigazolás
getAllRecipes	Összes recept lekérdezése	-	Receptek tömbje
deleteRecipe	Recept törlése admin jogosultsággal	recipeId: szám	Törlés visszaigazolás

#### Biztonság

- Minden függvény JWT token-t küld az Authorization fejlécben
- Az adminisztratív funkciók szerveroldali jogosultság-ellenőrzésen mennek keresztül

### 4. commentservice.js

#### Áttekintés

A commentservice.js modul a receptekhez tartozó hozzászólásokat kezeli, lehetővé téve a hozzászólások lekérdezését és új hozzászólások hozzáadását.

#### API végpont

- Comments API: <http://localhost:8080/api/comment>

Függvény	Leírás	Paraméterek	Visszatérési érték
getRecipeComments	Recepthez tartozó hozzászólások lekérése	recipeld: szám	Hozzászólások tömbje
addComment	Hozzászólás hozzáadása egy recepthez	recipeld: szám, text: szöveg	Létrehozott hozzászólás objektum

### Autentikáció

- A hozzászólások olvasása nem igényel autentikációt
- Új hozzászólás létrehozása JWT token autentikációt igényel

### Közös jellemzők

- Axios használata: Mindegyik service az axios HTTP klienst használja a backend API-val való kommunikációhoz
- JWT autentikáció: A védett végpontok a localStorage-ból kiolvasott JWT tokenet használják
- Hibakezelés: A hibák kezelése a service-eket használó komponensek felelőssége
- Aszinkron kérések: Minden API hívás Promise alapú és async/await szintaxissal van megvalósítva
- Adatformázás: A dátumok megfelelő formátumba való konvertálása (YYYY-MM-DD) a dayjs library segítségével történik

A service réteg biztosítja a tiszta elválasztást az API kommunikáció és a felhasználói felület között, megkönnyítve a kód karbantartását és esetleges API változtatások kezelését.