☕

Coffee App Documentation

Overview

The Angular Coffee App is a web application that allows users to search and filter a list of coffee products. The app provides a user-friendly interface for searching coffee products based on their names. The filtered results are displayed as cards, providing information about each coffee product. By default or when the search bar is empty, the whole list is displayed.

Components

The app consists of the following components:

1. Coffee Component

The `coffee.component` is responsible for rendering the coffee cards. It receives an array of `coffeeData` as an input and displays the coffee cards based on the filtered and ordered `coffeeData`. Each coffee card includes information such as the coffee name, image, rating, description, and price. Users can click on the **"+"** button to add a coffee to their cart. The `coffee.component` also emits the `filteredCoffeeList` to the parent component.

> The `addToCart()` method in the `coffee.component` is triggered when the user clicks the "+" button on a coffee card. Currently, it displays a simple alert to indicate that the coffee has been added to the cart. You can customize this method to perform your desired action, such as updating a shopping cart or managing a state variable.

2. Main Component

The `main.component` serves as the main container for the application. It handles the search functionality and communicates with the Coffee component. The Main component subscribes to the coffee data from the `coffee.service` and initializes the `filteredCoffeeList` and `searchedCoffeeList` arrays. It also provides the `getCoffeeData()` method to pass the `searchedCoffeeList` to the `coffee.component`. The `main.component` updates the `searchedCoffeeList` based on the user's search input, filtering the coffee data by name.

3. App Component

The `app.component` represents the main layout of the application. It includes the toolbar at the bottom, which consists of icons for home, shop, favorite, notification and notification-dot. The toolbar is fixed at the bottom of the screen and provides easy access to different sections of the app.

Logic

The Angular Coffee App implements the following logic:

Filtering Coffee Products

1. The `coffee.component` receives the list of coffee products and the filtered coffee list as inputs.
2. When the user enters search terms in the search bar and clicks the search button, the `main.component` updates the `searchTerm` property.
3. The `main.component` detects changes to the `searchTerm` property and triggers the `filterCoffeeList()` method.
4. The `main.component` renders the filtered coffee list by iterating over the `filteredCoffeeList` property to render the list.
5. The `filterCoffeeList()` method checks if a search term exists. If it does, it filters the coffee products based on the search term and updates the `filteredCoffeeList` property. The filtering logic uses the `Onsearch()` method in this way:

```
const searchTermLower = deburr(this.searchTerm.toLowerCase()); - This line creates a lower
case version of the searchTerm property and removes accents;
```

```
if (searchTermLower.trim() === '') { - This condition checks if searchTermLower is empty;

this.searchedCoffeeList = this.filteredCoffeeList; - If the search term is empty, all coff
ee items are included;

else { - If there is a search term, the code inside this block is executed;

this.searchedCoffeeList = this.filteredCoffeeList.filter((coffee) => { - The searchedCoffe
eList property is a filtered version of the filteredCoffeeList property;

(coffee) => { - This is the callback function for filtering each coffee item;

const coffeeTitleLower = deburr(coffee.name.toLowerCase()); - This line creates a lowercas
e version of the name property of the current coffee item;

return (coffeeTitleLower.normalize('NFD').replace(/[\\\\u0300-\\\\u036f]/g, '').includes(s
earchTermLower) || this.containsPartialWords(coffeeTitleLower, searchTermLower)); - This l
ine determines whether a coffee item should be included in the filtered array based on whe
ther the normalized coffeeTitleLower (removing accents) includes the searchTermLower or if
the containsPartialWords() function returns true;

The onSearch() method filters coffee items based on a search term.
```

> To be clear, I named the list of coffees that I retrieved from the API call `filteredCoffeeList` because I've chosen to filter some coffees. This because their photos did not match the others. I then sliced the filtered list to 14 items to avoid displaying products that were not actual drinks (such as coffee powders). I chose only the most relevant coffees to display.

Resetting the Filter

1. When the user wants to reset the filter and display all coffee products, they can just clear the search bar.
2. The `main.component` detects the click event and triggers the `OnSearchInput()` method.
3. The `OnSearchInput()` method clears the `searchTerm` property and sets the `searchedCoffeeList` property back to the original `filteredCoffeeList`, displaying all coffee products.

Usage

To use the Angular Coffee App in your project, follow these steps:

1. Install Angular if you haven't already:

   ```
   bash
   npm install -g @angular/cli
   ```

2. Install Lodash to have the accent insensitive search, font-awesome to have the icons and the search bar:

   ```
   npm install lodash
   npm install @fortawesome/fontawesome-free
   ```

3. Clone the repository or copy the relevant files into your Angular project.
4. Import the necessary components (`coffee.component`, `main.component`, `app.component`) into your module and add them to the desired templates.
5. Update the data source for coffee products (`coffeeList`) as per your requirements. Ensure that each coffee product object has properties like `name`, `description`, and `image`.
6. Customize the styling and UI of the components as needed.
7. Build and run your Angular project (ng serve inside at least the coffeeapp folder):

   ```
   bash
   ng serve
   ```

8. Access the app in your browser at `http://localhost:4200` and start searching and filtering coffee products.

This documentation provides an overview of the Angular Coffee App, its components, the logic behind filtering, and resetting the filter. It also includes instructions on how to use the app in your project. Feel free to modify and expand upon this documentation based on your specific implementation details.