Angular Coding Challenge - Recipe app

Let's build a webapp to help users cook a meal from the ingredients they have in their fridge.

The idea is to display a list of recipes and a list of ingredients. When users select the ingredients they have in their fridge, the app will highlight what recipe they can cook and what ingredients they would need to shop.

You can find attached a JSON list of recipes as well as a JSON list of ingredients: use this data in your application.

Base requirements from a product perspective

- Display a list of recipes. Each recipe will also show the ingredients contained in that recipe.

  Example: Burger: bun, ham, cheese || Pizza: flour, tomato sauce, cheese

- Display a SEPARATE list of all available unique ingredients, elements are selectable, they will be deselected by default.

  Example: bun, ham, cheese, flour, tomato sauce

- When selecting an ingredient, we want to show to the user which recipe they can go for and which ingredients they are missing:

  1. All recipes containing that ingredient will be highlighted.
  2. Selected ingredients within these recipes will be highlighted as well.

     Example 1: **Burger**: bun, **ham**, cheese || Pizza: flour, tomato sauce, cheese

     bun, **ham**, cheese, flour, tomato sauce

     Example 2: **Burger**: bun, ham, **cheese** || **Pizza**: flour, tomato sauce, **cheese**

     bun, ham, **cheese**, flour, tomato sauce

- When a recipe has ALL its ingredients selected: highlight it in a separate way to show that the user has all the ingredients at their disposal to go for this recipe.

Base requirements from a technical perspective:

- Use Angular 17+.
- Use NgRx and RxJS.
- Please ship either a zip or a link to a repo, make sure everything is included.
- You can use whatever additional package you want.
- You can ask ChatGPT/Copilot for help if you can explain what you ship in the end.
- We do not expect outstanding UI work, but it would be nice if it looked appealing.

Notes/Tips

- In practice NgRx is not needed for a simple exercise such as this, but we are interested in seeing how you use NgRx and RxJS.
- Be mindful of how you define the state of the app.
- Using @ngrx/effects would be appreciated.
- Strict typings would be appreciated.
- A separation between global-state and view-model would be appreciated.
- Good action hygiene would be appreciated.
- Handling API errors would be appreciated.
- Writing ONE unit test for a service and ONE unit test for a component would be appreciated.
- Component-store, signals, signal-store: feel free to use them if you want to.

Bonus Features

These are not compulsory; you can go for them if you want to! It would be greatly appreciated, and it is a good occasion to show off your expertise!

Bonus Feature 1 - Saving selected ingredients

- Use @ngrx/effects to trigger a fake API call when selecting an ingredient to save it in a back end.
    - Use local storage instead of a back-end solution.
- Hydrate the list of selected ingredients from this fake API on page load.
- Handle loading state (add a 300ms delay on the fake API call to simulate slow network).
- Handle errors.

Bonus Feature 2 - Cart checkout

- Allow users to click on a recipe to add the missing ingredients to a cart.
- Allow users to check out the cart (no payment process) and trigger a fake API call.
- Make this API call fail 10% of the time and handle errors.

Evaluation Criteria

- Angular best practices.
- NgRx best practices.
- Knowledge of RxJS and observables.
- Code readability.
- TypeScript typings.

ff3e568a272ca876d331675fd3d5a6c2