

Designing a lab exercise focused on Angular's structural directives `*ngFor` and `*ngSwitch` provides an excellent opportunity for learners to practice iterating over data collections and handling multiple conditions within templates. This exercise will guide you through creating a simple Angular application that demonstrates these directives by building a dynamic list with categorized items.

Lab Exercise: Mastering `*ngFor` and `*ngSwitch` in Angular

Objective

Learn how to use the `*ngFor` directive to iterate over data collections and the `*ngSwitch` directive to handle multiple conditions within Angular templates.

Requirements

- Angular CLI installed on your machine.
- Basic understanding of Angular application structure.
- Familiarity with TypeScript and HTML.

Setup

1. Create a new Angular project: `ng new ngForNgSwitchPractice`.
2. Navigate into your project directory: `cd ngForNgSwitchPractice`.
3. Serve your application to verify its setup: `ng serve`.
4. Open the project in your preferred code editor.

Exercises

Exercise 1: Iterating with `*ngFor`

1. In the app component (`app.component.ts`), create an array property named `items` filled with objects. Each object should represent an item with properties like `id`, `name`, and `category`.
2. Use the `*ngFor` directive in the app component template (`app.component.html`) to display each item in a list (``). Display the `name` of each item.

Exercise 2: Categorizing Items with `*ngSwitch`

1. Modify the `items` array to include items from at least three different categories.
2. Above or below the list, add a segment of HTML to display a message about the category of each item as you iterate over them. Use the `*ngFor` directive to iterate over the items and the `*ngSwitch` directive on a container element (`<div>` or ``) to display a different message or HTML structure for each category.

Exercise 3: Filtering Items

1. Add an input field above the list to filter items by name.
2. Implement a method in the app component that filters the displayed items based on the user's input. This might involve creating a new array for the filtered items or adjusting the `*ngFor` directive to use a pipe for filtering.
3. Ensure that the list dynamically updates as the user types in the filter input field.

Exercise 4: Adding and Removing Items

1. Add functionality to add new items to the list. Include input fields for the `name` and `category` of the new item and a button to add the item to the list.
2. Implement a method in the app component that adds a new item to the `items` array based on the input field values.
3. Add a button next to each item in the list that removes the item from the list when clicked. Implement the removal logic in the app component.

Conclusion

By completing this lab exercise, you will have practiced using Angular's `*ngFor` directive to iterate over arrays and display lists dynamically. Additionally, you will have explored using the `*ngSwitch` directive to conditionally display content based on the value of an expression, specifically for categorizing items. This exercise also touches on dynamic list manipulation, including adding and removing items