Week 10: Angular Fundamentals & Messenger Frontend Setup

FH University of Applied Sciences
TECHNIKUM WIEN

Week 10: Angular Fundamentals & Messenger Frontend Setup

In this session, we will:

- Introduce Angular's core concepts.
- Learn how to install & configure Angular.
- Explore components, modules, and basic data binding.
- Start building a modular messenger frontend.

1. Why Angular?

- **TypeScript-based**: Angular is built with TypeScript, offering type safety.
- **Component-Driven**: Encourages modular architecture.
- **Powerful CLI**: Simplifies project creation, code generation, and testing.
- **Robust Ecosystem**: Numerous libraries, strong community support.

Real-World Use Cases

- Large enterprise SPAs
- Scalable and maintainable projects

2. Getting Started with Angular

Installation & Setup

1. **Node.js & npm**: Ensure Node.js is installed (`node -v`).
2. **Angular CLI**: `npm install -g @angular/cli`
3. **Editor Extensions**: Visual Studio Code with Angular Snippets, ESLint/TSLint, Prettier.

Creating a Project

- `ng new messenger-frontend`
- Generated folders:
  - **`app/`**: Root module & components
  - **`environments/`**: Environment configs
  - **`main.ts`**: Application entry point

Serving the App

- `cd messenger-frontend`
- `ng serve` -> Open `http://localhost:4200`

3. Angular Core Concepts

Modules

- Group related components, services, pipes.
- **Root Module** (`AppModule`) is bootstrapped on startup.

Components

- The building blocks of an Angular app.
- Decorator (`@Component`) with `selector` , `template` , and `styles` .

Data Binding

1. **Interpolation**: `{{ variable }}`
2. **Property Binding**: `[property]="expression"`
3. **Event Binding**: `(event)="handler()"`
4. **Two-Way Binding**: `[(ngModel)]="property"` (requires `FormsModule`)

Code Demonstration

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-welcome',
  template: `<h2>Welcome to Angular Messenger!</h2>`,
  styles: [ 'h2 { color: blue; }' ]
})
export class WelcomeComponent {
}
```

4. Hands-On Practice

1. **Create & Serve an Angular Project**
   - Install Angular CLI.
   - Run `ng new messenger-frontend`.
   - `ng serve` to see your app.
2. **Generate a Component**
   - `ng generate component intro`
   - Add a welcome message in `intro.component.html`
   - Display the new component in `app.component.html`

Week 10 Exercise

Overview

We'll begin building the **messenger frontend**, focusing on:

- Basic Angular project structure.
- A modular approach for future expansions.

1. Set Up Angular

- Create a new project: `ng new messenger-frontend`
- Verify it runs with `ng serve`.

2. Build a Feature Module

- Generate a `MessageModule`: `ng generate module message`
- Inside `MessageModule`, create a `MessageListComponent` with mock messages.

3. Optional: Create an Intro Component

- Welcome users.
- Practice data binding.

Exercise Sample Solution

`messenger-frontend/`

- `app/`: Root module & component.
- `message/`: New feature module.
  - `message-list.component.ts` to display mock messages.

Example:

```
// message-list.component.ts
@Component({
  selector: 'app-message-list',
  template: `
    <div *ngFor="let msg of messages">
      <strong>{{ msg.sender }}</strong>: {{ msg.content }}
    </div>
  `,
})
export class MessageListComponent {
  messages = [
    { sender: 'Alice', content: 'Hello from Angular!' },
    { sender: 'Bob', content: 'Can't wait to connect to a backend!' }
  ];
}
```

Summary & Next Steps

- **Week 10:** Angular Fundamentals & Project Setup
- **Week 11:** Services & RESTful Integration (send/fetch messages)
- **Week 12:** Routing, optional real-time features
- **Week 13:** Final best practices & finishing touches

Keep exploring Angular’s documentation and experiment with components.

Happy coding!