

Angular Guide: From Beginner to Intermediate

Table of Contents

- [Introduction to Angular](#)
- [Setup and Environment](#)
- [Core Concepts](#)
- [Servicesand Dependency Injection](#)
- [Routing and Navigation](#)
- [Forms in Angular](#)
- [HTTPClientand APIs](#)
- [Styling and Theming](#)
- [Testingin Angular](#)
- [Mini Projects & Examples](#)
- [Resources& Cheat Sheets](#)
- [Further Learning](#)

1. Introduction to Angular

Angular is a platform and framework for building single-page client applications using HTML and TypeScript

Angular vs Reactvs Vue: Angular is a full-fledged framework, Reactis a library, and Vue is a progressive framework.

History and evolution : AngularJSwas the first version, followed by Angular 2+, which is a complete rewrite.

Keyfeatures include MVVM, CLI, and Dependency Injection.

2. Setup and Environment

Installing Node.js & Angular CLIUse npm to install the Angular CLIglobally.

Creating your first Angular project: Use the CLIcommand `ng new project-name`.

Folder structure explanation: Understand the src, app, and assetsdirectories.

3. Core Concepts

Modules: Angular apps are modular and use NgModules to organize code.

Components: Use decorators, metadata, and templates to define components.

Templatesand Data Binding: Bind data from component to template using interpolation and property binding.

Directives: Use built-in directives like ngIf and ngFor, or create custom directives.

Pipes: Transformdata in templates using built-in and custom pipes.

4. Services and Dependency Injection

Creating and injecting services Use Angular's DI system to inject servicesinto components.

Singleton behavior: Servicesare singletons by default, providing a single instance across the app.

5. Routing and Navigation

Setting up routes: Define routes in the app-routing.module.ts file.

Route parameters: Passparameters to routes for dynamic content.

Lazyloading modules: Load modules on demand to improve performance.

6. Forms in Angular

Template-driven forms: Use Angular directives to create forms in templates.

Reactiveforms: Use FormBuilder to create forms programmatically.

Form validation and error messages Validate forms and display error messages

7. HTTPClient and APIs

Making API calls: UseHttpClient to make HTTPrequests to APIs

Observableswith RxJSHandle asynchronous data streams with Observables

Error handling: Manage errors in HTTPrequests using catchError.

8. Styling and Theming

Component styles: Stylecomponents using CSSor SCSS

Global styles: Define global styles in styles.cssor styles.scss

Angular Material introduction : Use Angular Material for UI components.

9. Testing in Angular

Unit testing with Jasmine& Karma: Write unit tests for components and services

E2Etesting with Protractor: Testthe entire application flow.

10. Mini Projects & Examples

Todo App: Create a simple todo application.

Weather App using API: Build a weather app using a public API.

Simple CRUDwith Firebase Implement CRUDoperations using Firebase

11 Resources& Cheat Sheets

Angular Official Docs: Comprehensive documentation for Angular.

Angular Material UI: Explore UI components for Angular.

Roadmap links: Follow the Angular roadmap for future updates.

GitHub repositories: Explore open-source Angular projects on GitHub.

Further Learning

Explore videos, blogs, and documentation for further learning.

