Angular A to Z Guide (Basic to Advanced)

==============================

Angular A to Z Guide

==============================

A. Angular Basics

- What is Angular?

- Angular vs AngularJS

- Setting up Angular CLI

- Project Structure Overview

- Components, Modules, Services

- Templates and Data Binding

B. Components and Modules

- Creating and using components

- Component communication (@Input, @Output)

- Creating and using modules (lazy loading)

C. Directives and Pipes

- Built-in directives (ngIf, ngFor, etc.)

- Custom directives

- Built-in pipes (date, currency, etc.)

- Custom pipes

D. Routing and Navigation

- RouterModule and AppRoutingModule

- Route parameters and guards

- Lazy loading routes

E. Forms in Angular

- Template-driven forms

- Reactive forms

- Form validation (sync/async)

- Dynamic form generation

F. Services and Dependency Injection

- Creating services

- Injectable decorator

- HTTP requests using HttpClient

- Observables and RxJS basics

G. Authentication & Authorization

- Register/Login UI & logic

- AuthGuard to protect routes

- JWT token-based auth

- Token Interceptor for secure APIs

H. Angular Material UI

- Installing Angular Material

- Material components (button, input, table, etc.)

- Dialogs, snackbars, tooltips

I. State Management

- @Input/@Output vs Service pattern
- RxJS BehaviorSubject
- Introduction to NgRx (for larger apps)

J. Advanced Topics

- Change Detection strategies
- ViewChild & ContentChild
- Custom Events and Global Events
- Angular Animations

K. Unit Testing

- Testing components with Jasmine & Karma
- Testing services and pipes

L. Performance Optimization

- Lazy loading modules
- OnPush change detection
- Code splitting and preloading strategies

M. Deployment

- Building for production
- Environment files (dev vs prod)
- Deploying to Firebase, Netlify, or Vercel

N. Best Practices

- Folder structure

- Naming conventions
- Clean code and reusable components

O. Tools & Resources

- Angular DevTools
- StackBlitz, Plunker for online prototyping
- Official Angular Docs

==============================

Bonus: Working in a Company

==============================

- Git Workflow (pull, push, branches, merge)
- Agile methodology (sprints, standups)
- Code reviews and Pull Requests (PRs)
- Working with APIs and backend integration (.NET, Node.js)
- Using Postman to test APIs
- Jira/Trello for task management
- Communication with team (Slack, Email, Meetings)

Tip: Always write reusable, testable, and maintainable code.

Learn Build Test Deploy Repeat