

ANGULAR TRAINING

-Darathi J

What is Angular ?

-
- Web Framework that helps in building reliable applications
 - Maintained by Google
 - Opensource and JavaScript Framework
 - It includes tools, libraries for building ,testing and maintaining .
 - Cross Platform Support
 - Custom and Reusable components

Why Angular ?

-
- Easier maintenance of code
 - Lightweight architecture contributes faster loading times and user experience
 - Large and Active community support
 - Scalability -It can scale from single-developer projects to enterprise-level applications.
 - Better error handling
 - Supports multiple frameworks for f

System Setup

- Install Node.js(Angular 15 requires Node.js version **14.20.x**, **16.13.x**, or **18.10.x.....**)
- To check if node is properly:

```
C:\Users\jdrathi>node --version  
v22.13.0
```

```
C:\Users\jdrathi>npm -v  
11.0.0
```

```
C:\Users\jdrathi>node -v  
v22.13.0
```

Install Angular

- To install Angular CLI globally
- `npm install -g @angular/cli@15`
- To check proper installation
- `ng version`

```
C:\Users\jdrathi>ng version
```

Angular CLI

```
Angular CLI: 19.1.0  
Node: 22.13.0  
Package Manager: npm 11.0.0  
OS: win32 x64
```

```
Angular:  
...
```

Package	Version
@angular-devkit/architect	0.1901.0 (cli-only)
@angular-devkit/core	19.1.0 (cli-only)
@angular-devkit/schematics	19.1.0 (cli-only)
@schematics/angular	19.1.0 (cli-only)

Create a new Project

-
- `ng new my-study-app`
 - You'll be prompted to:
 - Add Angular routing? → **Yes/No**
 - Select stylesheet format (CSS/SCSS/etc.)
- It will automatically install dependencies.

```
C:\Users\jdrathi>ng new my-study-app
✓ Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS
✓ Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? Yes
✓ Would you like to use the Server Routing and App Engine APIs (Developer Preview) for this server application? Yes
CREATE my-study-app/angular.json (2870 bytes)
CREATE my-study-app/package.json (1281 bytes)
CREATE my-study-app/README.md (1532 bytes)
CREATE my-study-app/tsconfig.json (942 bytes)
CREATE my-study-app/.editorconfig (331 bytes)
CREATE my-study-app/.gitignore (629 bytes)
CREATE my-study-app/tsconfig.app.json (508 bytes)
CREATE my-study-app/tsconfig.spec.json (449 bytes)
CREATE my-study-app/.vscode/extensions.json (134 bytes)
CREATE my-study-app/.vscode/launch.json (490 bytes)
CREATE my-study-app/.vscode/tasks.json (980 bytes)
CREATE my-study-app/src/main.ts (256 bytes)
CREATE my-study-app/src/index.html (309 bytes)
CREATE my-study-app/src/styles.css (81 bytes)
CREATE my-study-app/src/main.server.ts (271 bytes)
CREATE my-study-app/src/server.ts (1674 bytes)
CREATE my-study-app/src/app/app.component.html (20239 bytes)
CREATE my-study-app/src/app/app.component.spec.ts (963 bytes)
CREATE my-study-app/src/app/app.component.ts (300 bytes)
CREATE my-study-app/src/app/app.component.css (0 bytes)
CREATE my-study-app/src/app/app.config.ts (447 bytes)
CREATE my-study-app/src/app/app.routes.ts (80 bytes)
CREATE my-study-app/src/app/app.config.server.ts (519 bytes)
CREATE my-study-app/src/app/app.routes.server.ts (174 bytes)
CREATE my-study-app/public/favicon.ico (15086 bytes)
\ Installing packages (npm)...
```

Serve the Application

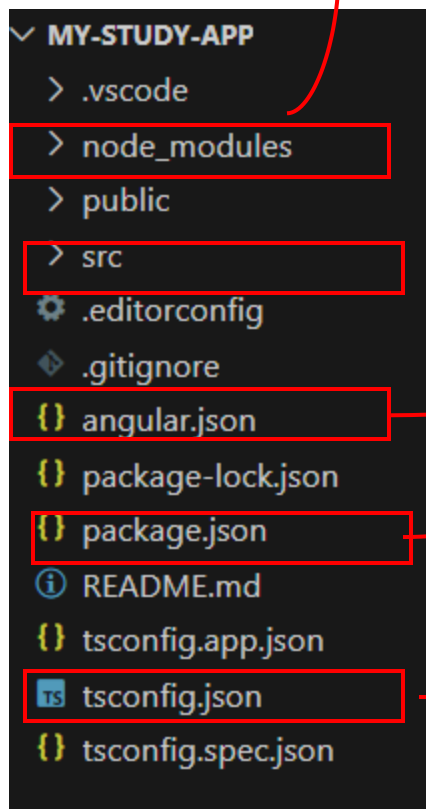
-
- `cd my-study-app`
 - `ng serve`

Then open <http://localhost:4200> in your browser.

The image features a large, thin white circle centered on a black background. A thick, light green line follows the inner curve of the bottom half of this circle. To the left of the circle, there are two horizontal wavy lines. Below them is a small, solid light orange circle. In the top right corner, there is a small double-circle outline in light orange. In the bottom right corner, there is a rectangular grid of small white dots.

FOLDER STRUCTURE

Contains all the **installed dependencies** (libraries) like Angular, RxJS, etc.



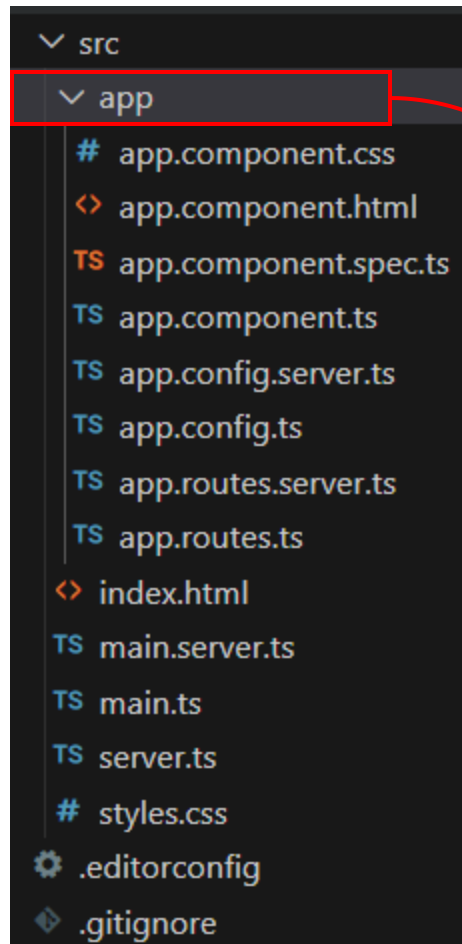
Heart of your application, whatever we see in the browser is inside this folder

Main config file for your Angular project.

- Tells Angular how to build and serve the app.
- Manages styles, scripts, and build settings.

- Lists all the **npm packages** your app uses.
- Defines the scripts to build and serve the app.

Configuration file for **TypeScript** — sets rules like how strict the code should be, and how modules are handled.



This folder contains your **Angular components, services, and modules**.

- app.component.ts: Main component with your logic (TypeScript)
- app.component.html: UI template
- app.component.css: Styles for the component
- app.module.ts: Registers components and services

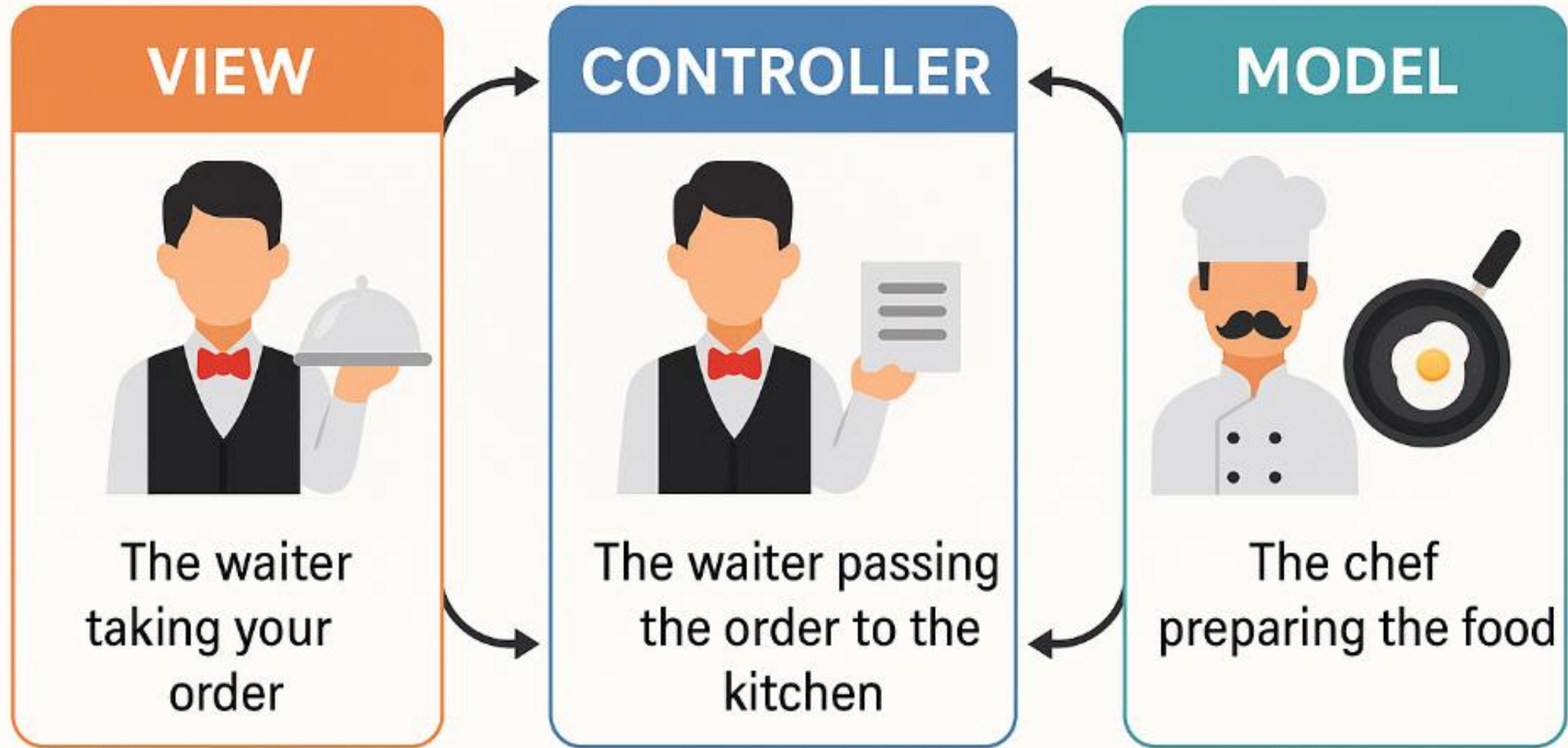
assets/

Used for **static files** like images, icons, fonts, etc.

A graphic with a black background. A large white circle is centered, with a thick green line tracing its inner edge. Inside the circle, the text 'MVC ARCHITECTURE' is written in white, bold, sans-serif font. To the left of the circle, there are two white wavy lines. Below them is a small solid orange circle. To the right of the circle, there is a small double-lined orange circle in the top right corner and a rectangular grid of small white dots in the bottom right corner.

MVC ARCHITECTURE

MVC ARCHITECTURE



-
- MVC stands for:
Model
View
Controller
 - It's a **design pattern** used to organize code so that it's easier to manage and scale.
-

Part	Role	Angular Equivalent
Model	Represents the data and business logic	Interfaces, services, APIs
View	What the user sees (UI)	HTML templates (.html)
Controller	Manages user input, updates the model/view	Component class (.ts files)

The image features a central white circle with a thick green border. Inside this circle, the word "COMPONENTS" is written in bold, white, uppercase letters. Surrounding the central circle are several abstract elements: two white wavy lines to the upper left, a small orange circle to the lower left, a small orange ring to the upper right, and a grid of white dots to the lower right.

COMPONENTS

What is Component?

- Components are the main building blocks for Angular applications. Each component consists of:
- An HTML template that declares what renders on the page
- A TypeScript class that defines behavior
- A CSS selector that defines how the component is used in a template
- Optionally, CSS styles applied to the template
- Creating a component : `ng generate component <component-name>`
- By default this command creates :
 - A directory named after the component
 - A component file, `<component-name>.component.ts`
 - A template file, `<component-name>.component.html`
 - A CSS file, `<component-name>.component.css`
 - A testing specification file, `<component-name>.component.spec.ts`

Creating a Component

- Creating a component : `ng generate component <component-name>`
- By default this command creates :
 - A directory named after the component
 - A component file, `<component-name>.component.ts`
 - A template file, `<component-name>.component.html`
 - A CSS file, `<component-name>.component.css`
 - A testing specification file, `<component-name>.component.spec.ts`