Angular | Lecture 1

Marina Magdy

Agenda

- What is angular ?
- What is the alternates to angular?
- Single Page Application
- Getting started with angular
- Angular building blocks : Components.
- Templates and styles
- Data Binding
- Lifecycle Methods

What is Angular ?

- Angular is a **Javascript framework** which allows you to create single page applications. It's only one HTML file and a bunch of JavaScript code we got from the server that changes content of this HTML.
- Angular is using **Typescript** which is a superset of javascript and compiles to Javascript .
- Angular current stable version is **17.3.0**
- Angular was developed by Google team
- Angular docs : https://angular.io/docs

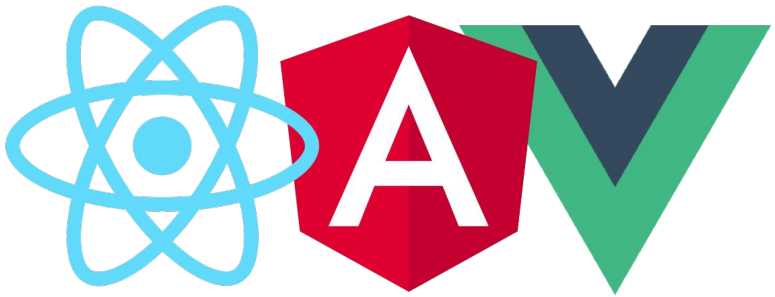
What is Angular ?

- A component-based framework for building scalable web applications
- A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more
- A suite of developer tools to help you develop, build, test, and update your code

Angular Alternates

Stack overflow Survey :

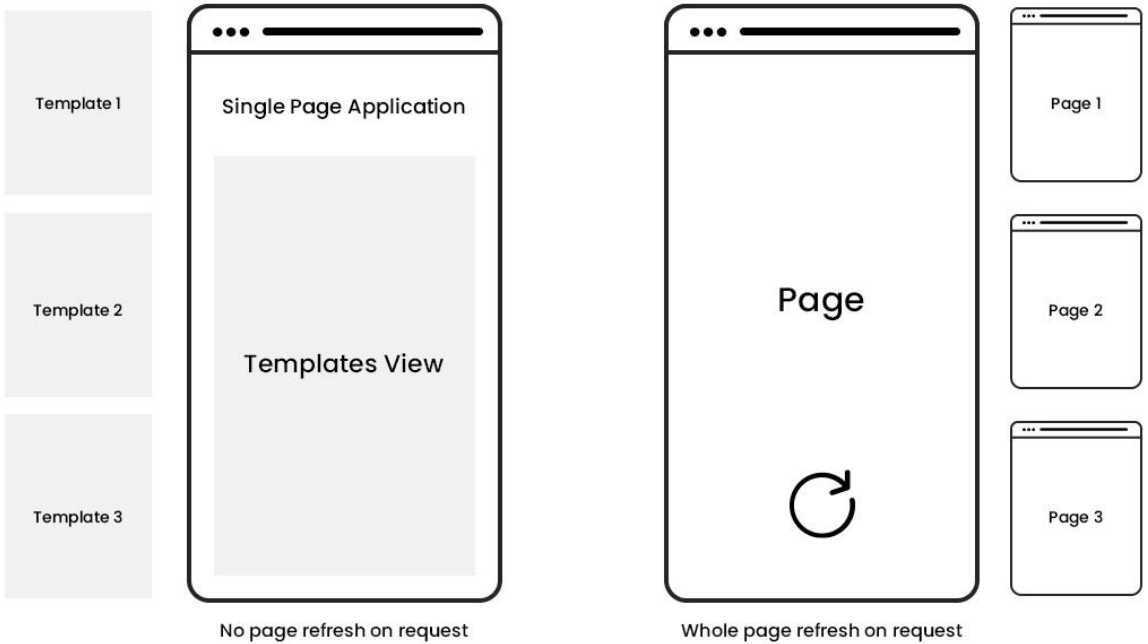
https://survey.stackoverflow.co/2023/#most-popular-technologies-webframe



Single page application

- Single Page Applications (SPAs) all the HTML generation happens in the browser. The server only returns one basic HTML page for all incoming requests (no matter the URL).
- But that single HTML page contains a lot of JavaScript code (typically outsourced into separate files) which is responsible for changing the HTML code (technically, the DOM).
- Single page application examples : gmail,facebook,netflix ...etc

Single page application



Angular : Getting Started

- Install npm (node package manager) :

  https://nodejs.org/en/

  What is NPM ?

  Node Package Manager (NPM) is a command line tool that installs, updates or uninstalls Node.js packages in your application. It is also an online repository for open-source Node.js packages. The node community around the world creates useful modules and publishes them as packages in this repository.

Angular : Getting Started

- Install Angular CLI (command line interface) globally :

  npm install -g @angular/cli

  What is CLI ?

  The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

  Then try to run **ng version** to make sure that angular/cli installed.

Angular : Getting Started

To create your first angular app , run the following command in CMD :

- **ng new project-name** (create angular app)
- **cd ./project-name** (enter project folder)
- **ng serve -o** (run application and -o (--open) refers to open application automatically in browser)

Angular : Getting Started

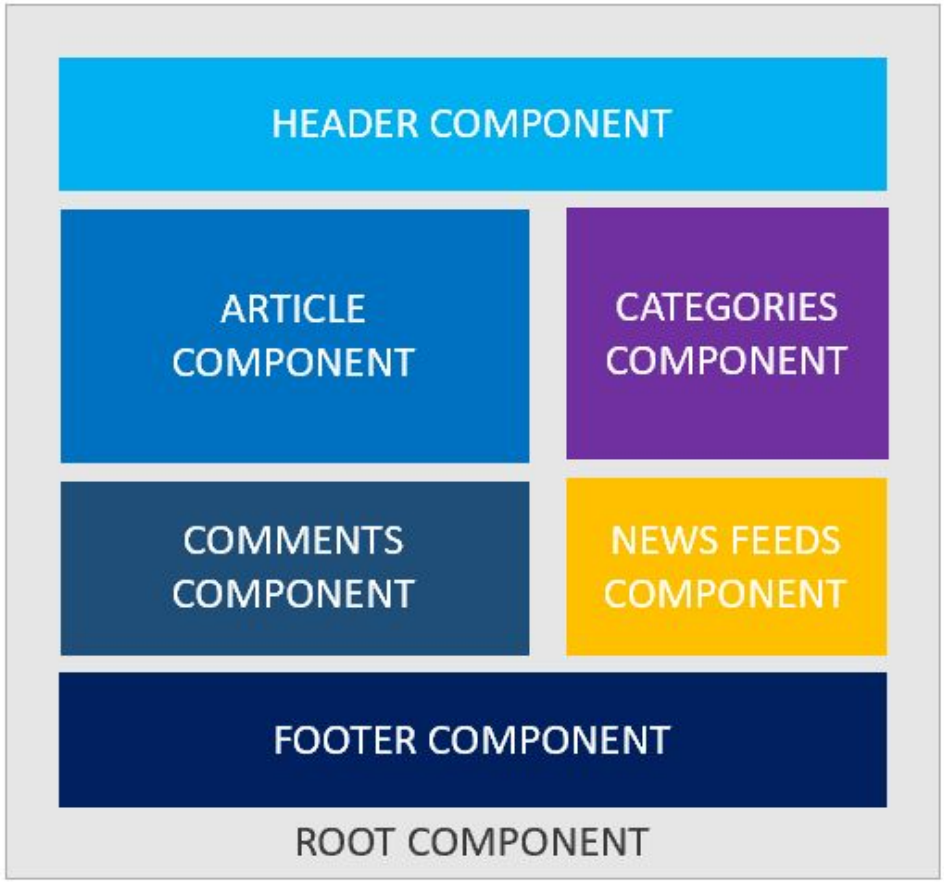
Let’s discover and have a deep look at our angular app structure

https://angular.io/guide/file-structure

https://angular.dev/reference/configs/file-structure#workspace-configuration-files

Components

Components


HEADER COMPONENT
ARTICLE COMPONENT
CATEGORIES COMPONENT
COMMENTS COMPONENT
NEWS FEEDS COMPONENT
FOOTER COMPONENT
ROOT COMPONENT

Components

Components are the main building block for Angular applications , Components are composable, we can build larger Components from smaller ones. 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
- Spec file for testing

Standalone components

Components, directives, and pipes can now be marked as standalone: true. Angular classes marked as standalone do not need to be declared in an NgModule

Standalone components specify their dependencies directly instead of getting them through NgModules

Components

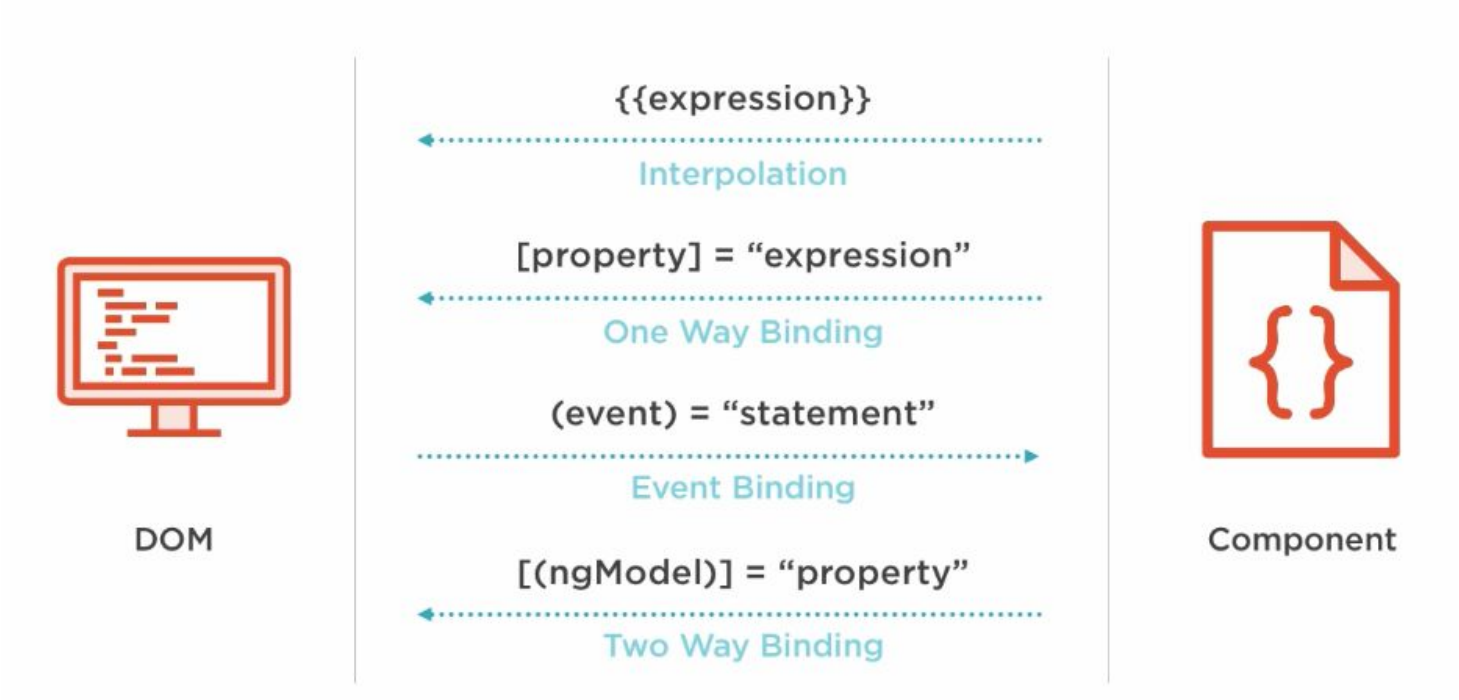
- **Create new component : ng generate component navbar**
- Discover new generated component class.
- @Component decorator: decorator that marks a class as an Angular component and provides configuration metadata that determines how the component should be processed, instantiated, and used at runtime.
- External template and styles

Reusable components

- To use component in any other place you can use it by selector name mentioned in @component between HTML tags.
- Create once , use multiple times.
- Syntax : **<app-navbar></app-navbar>**
- Add component to imports as it’s a standalone component

Data Binding

Data binding



Data binding

- **String Interpolation**

  Allows you to incorporate dynamic string values into your HTML templates

  used like this {{expression}}

- **Property Binding**

  Property binding in Angular helps you set values for properties of HTML elements or directives.

  Example : <img [src]="itemImageUrl">

Data binding

- **Event Binding**

  Event binding allows you to listen for and respond to user actions such as keystrokes, mouse movements, clicks, and touches.

  Example : <button (click)="onSave()">Save</button>

- **Two way binding**

  Two-way binding combines property binding with event binding for example to two way binding [(ngModel)]

  Note : need to import import { FormsModule } from '@angular/forms' for two way binding in forms in app module to work

Component lifecycle

Component lifecycle

- A component instance has a lifecycle that starts when Angular instantiates the component class and renders the component view along with its child views.
- You don't have to implement all (or any) of the lifecycle hooks, just the ones you need.
- After your application instantiates a component or directive by calling its constructor, Angular calls the hook methods you have implemented at the appropriate point in the lifecycle of that instance.

Component lifecycle

ngOnInit()

Initialize the directive or component after Angular first displays the data-bound properties and sets the directive or component input properties.

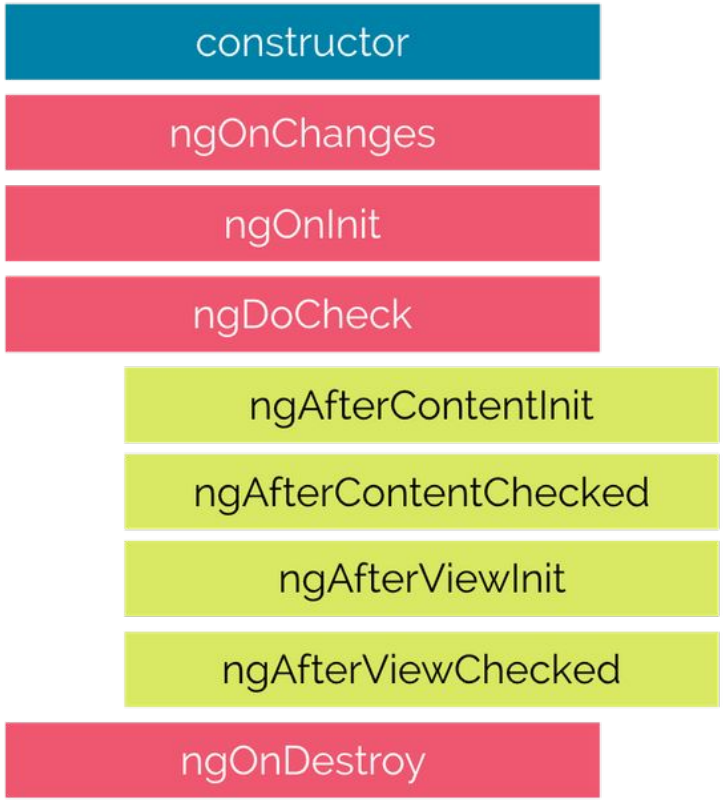
ngOnDestroy()

Cleanup just before Angular destroys the directive or component. Unsubscribe Observables and detach event handlers to avoid memory leaks.

ngAfterViewInit()

Respond after Angular initializes the component's views and child views, or the view that contains the directive.

Resources : https://angular.io/guide/lifecycle-hooks



Bootstrap

Bootstrap

Install Bootstrap : npm install bootstrap

Then add bootstrap css file to style.css

```
@import "../node_modules/bootstrap/dist/css/bootstrap.min.css"
```

you can also bootstrap angular component from **ngBootstrap** :

https://ng-bootstrap.github.io/#/getting-started

Install command : ng add @ng-bootstrap/ng-bootstrap

Thank you

Portfolio

Replace code found in app.component.html and start create your portfolio page using Bootstrap.

Portfolio will contain the following sections :

- Hero section (name and job title).
- Bio and about me (education and experiences) section with button to download CV .
- Skills section with progress bar for each skill
- Portfolio and projects section
- Footer contains contact us section with email and social media links (facebook , github , linkedin) [Will use fontawesome - **Bonus**]

Each section is a separate component.