04 Angular Fundamental

Prepared by Surajai Chamroensat

Agenda

- Overview of Angular
- Key features and benefits of Angular
- Setting up the development environment (Node.js, Angular CLI)
- Components
- Binding
- Handling Events
- Control Flow
- Input/Output
- Component Life Cycle
- Directive

Overview of Angular

Angular is a web framework that empowers developers to build fast, reliable applications.
Maintained by a dedicated team at Google, Angular provides a broad suite of tools, APIs, and libraries to simplify and streamline your development workflow. Angular gives you a solid platform on which to build fast, reliable applications that scale with both the size of your team and the size of your codebase.

Key features and benefits of Angular

- Typescript Language
- Support both client side / server side rendering
- Easy to learn
- Continuous development
- 2 Ways binding
- Active comunity

Setting up the development environment

- Node.js
  - https://nodejs.org/en/download
- Angular CLI
  - npm install -g @angular/cli
  - ng new <project-name>
  - npm start

Components

Components are the foundational building blocks for any Angular application. Each component has three parts:

- TypeScript class
- HTML template
- CSS styles

Binding

- Property Binding []
- Event Binding ()
- 2 Way Bindings [(ngModel)]

Handling Events

- click
- change

Controls Flow

- @if
- @for

Component with input/output property

Sometimes app development requires you to send data into a component or send the data from the component. This data can be used to customize a component or perhaps send information from a parent component to a child component and vice versa.

Component Life Cycle

Phase	Method	Summary
Creation	`constructor`	Standard JavaScript class constructor. Runs when Angular instantiates the component.
Change Detection	`ngOnInit`	Runs once after Angular has initialized all the component's inputs.
	`ngOnChanges`	Runs every time the component's inputs have changed.
	`ngDoCheck`	Runs every time this component is checked for changes.
	`ngAfterContentInit`	Runs once after the component's *content* has been initialized.
	`ngAfterContentChecked`	Runs every time this component content has been checked for changes.
	`ngAfterViewInit`	Runs once after the component's *view* has been initialized.
	`ngAfterViewChecked`	Runs every time the component's view has been checked for changes.
Rendering	`afterNextRender`	Runs once the next time that **all** components have been rendered to the DOM.
	`afterEveryRender`	Runs every time **all** components have been rendered to the DOM.
Destruction	`ngOnDestroy`	Runs once before the component is destroyed.

Directive

Directives are classes that add additional behavior to elements in your Angular applications.

Directive Types	Details
Components	Used with a template. This type of directive is the most common directive type.
Attribute directives	Change the appearance or behavior of an element, component, or another directive.
Structural directives	Change the DOM layout by adding and removing DOM elements.

Built-in Attribute Directive

Common directives	Details
`NgClass`	Adds and removes a set of CSS classes.
`NgStyle`	Adds and removes a set of HTML styles.
`NgModel`	Adds two-way data binding to an HTML form element.

References

- https://angular.dev/