Webontwikkeling 4

Angular

Elke Steegmans

Angular

- is a **complete JavaScript-based open-source front-end web application framework** mainly maintained by **Google** and by a community of individuals and corporations to address many of the challenges encountered in developing **single-page applications**

Versions

- AngularJS 1
- **Angular 2**

Angular 2

- is not a version upgrade, but a **complete rewrite**
- announced in **2014**
- is a **framework** for building client applications in **HTML** and either **JavaScript** or a language like **TypeScript** that compiles to JavaScript
- => called Angular from now on in the slides

Current version …

- Angular 9

Angular

- Angular applications are made up of **components**
  - A component is the combination of an **HTML template** and a **component class** that controls a portion of the screen

TypeScript

- Angular is mostly used in combination with TypeScript
    - is a superset of JavaScript
    - a real OO JavaScript :-)
    - TS code is compiled to JS code
- Angular can also be used in combination of JavaScript

Lectors Example

- https://github.com/UCLLWebontwikkeling4/Demo8_Angular_LectorExample_OnlyFrontEnd

Interpolation Binding

- The **double curly braces** are Angular's interpolation binding syntax.
  - These **interpolation bindings** present the component's title and hero property values, as strings, inside the HTML header tags.
- Also known as **one-way binding**

Classes and objects

- In TypeScript you can make **classes** and **objects** …

Template literals

- The **backticks** around the component template let you put the <h1>, <h2>, and <div> elements on their own lines, thanks to the **template literals** feature in ES2015 and TypeScript.

Lectors Example

- Users should be able to edit the hero name in an <input> text box. The text box should both display the hero's name property and update that property as the user types.

- You need a two-way binding between the <input> form element and the hero.name property.

Two-way Binding

- **[(ngModel)]** is the Angular syntax to bind the hero.name property to the textbox. Data flows in both directions: from the property to the textbox, and from the textbox back to the property = **two-way binding**.

- Although NgModel is a valid Angular directive, it isn't available by default. It belongs to the optional **FormsModule**. You must opt-in to using that module.

ngFor

- The built-in directive *ngFor
  - The (*) prefix to ngFor is a critical part of this syntax. It indicates that the <li> element and its children constitute a master template.
  - The ngFor directive iterates over the component's heroes array and renders an instance of this template for each hero in that array.

Building blocks of an Angular Application

- Module
- Component
- Directive
- Data binding

Module

- is a container for a group of related components, services, directives, and so on
- @NgModule
- all apps must have at least a root module that is bootstrapped during the app launch

AppModule

- Every component must be declared in one—and only one—Angular module.
    - AppModule
        - import
        - declarations

Component

- is the main building block of an Angular application
- each component consists of 2 parts
    - a view the defines user interface
    - a class that implements the logic behind the view
- @Component
- each app must have at least one component called the root component

Component

- each @Component must have
  - selector: is similar to a CSS selector
  - template: contains HTML markup

Directives

- allows you to attach custom behaviour to an HTML element
    - ngFor for example

Data Binding

- allows you to keep a component’s properties in sync with the view