

Web Frameworks

Angular – deel 1

Elektronica – ICT

Sven Mariën

(sven.marien01@ap.be)

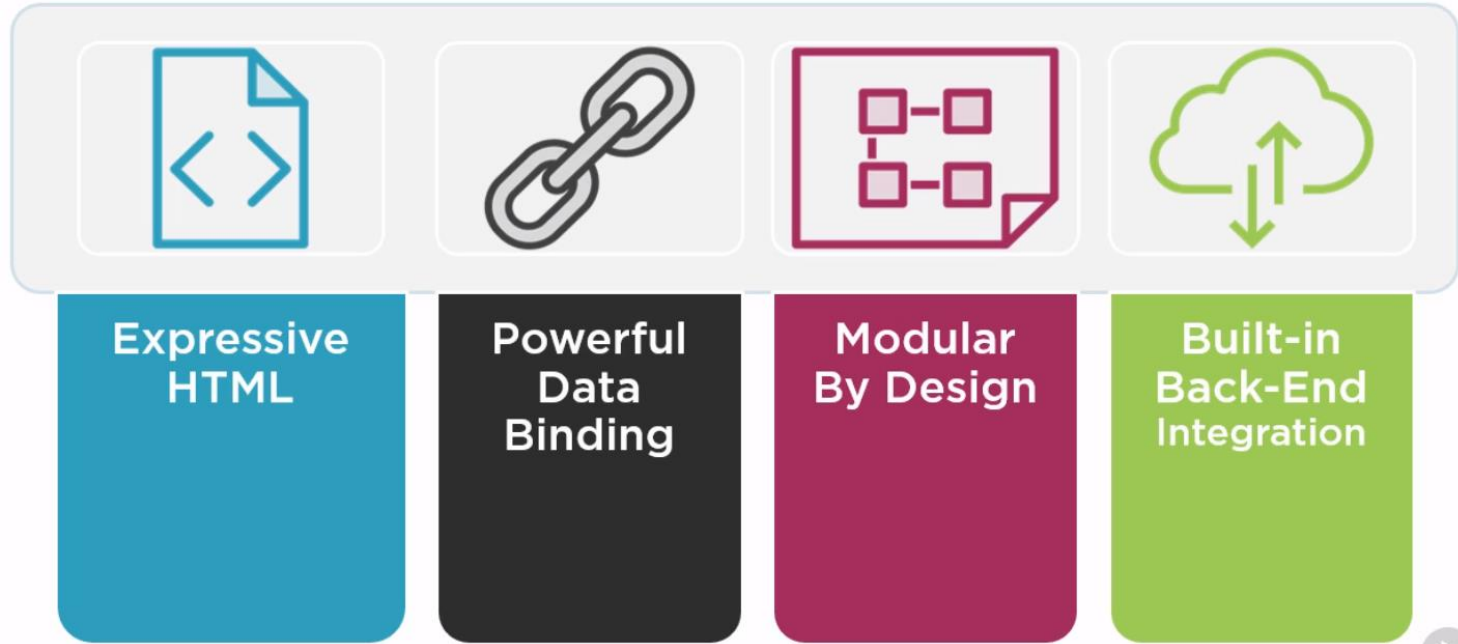
Inhoud

1. Deel 1

1. **Waarom Angular ?**
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - Van een Applicatie
 - Van een Component
4. Nieuwe Component Aanmaken

1.1. Waarom Angular ?

Why Angular?



Inhoud

1. Deel 1

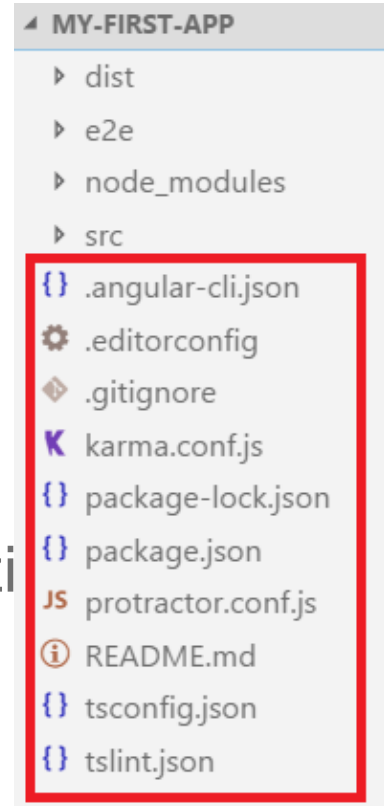
1. Waarom Angular ?
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - Van een Applicatie
 - Van een Component
4. Nieuwe Component Aanmaken

2.1. Nieuw Angular project aanmaken

- Gebruik van @angular-cli
 - Command line tool
 - Zorgt dat alle nodige bestanden aanwezig zijn voor een 'hello world' toepassing
 - Eenmalig (globaal) installeren via npm: **npm install -g @angular/cli**
 - Nieuw project aanmaken:
 - Navigeer naar de juiste map waar het project moet worden aangemaakt
 - **ng new PROJECT-NAAM** (maakt een nieuwe submap aan)
 - Druk telkens [enter] bij de vragen
 - **cd PROJECT-NAAM**
 - **Code .** (opstarten vs code)
 - **Npm start** (build & start een test webserver op die applicatie host)

2.2 Structuur van een project

- **'Node_modules'** map (npm modules, zie package.json)
- **'src'** map (onze eigen gemaakte bestanden project)
- **Configuratie** bestanden (package.json, tsconfig.json,...)
- **'Dist'** map (gecompileerd javascript, enkel zi een build)



package.json en dependencies

- Runtime:
 - @angular packages zijn opgedeeld volgens functionaliteit:

Feature Packages

@angular/core: Critical runtime parts of the framework needed by every application. Includes all metadata decorators, `Component`, `Directive`, dependency injection, and the component lifecycle hooks.

@angular/common: The commonly needed services, pipes, and directives provided by the Angular team.

@angular/compiler: Angular's *Template Compiler*. It understands templates and can convert them to code that makes the application run and render. Typically you don't interact with the compiler directly; rather, you use it indirectly via `platform-browser-dynamic` or the offline template compiler.

@angular/platform-browser: Everything DOM and browser related, especially the pieces that help render into the DOM. This package also includes the `bootstrapStatic()` method for bootstrapping applications for production builds that pre-compile templates offline.

@angular/platform-browser-dynamic: Includes `Providers` and a `bootstrap` method for applications that compile templates on the client. Don't use offline compilation. Use this package for bootstrapping during development and for bootstrapping plunker samples.

@angular/http: Angular's HTTP client.

@angular/router: Component router.

@angular/upgrade: Set of utilities for upgrading AngularJS applications to Angular.

```
package.json x
1  {
2    "name": "my-first-app",
3    "version": "0.0.0",
4    "license": "MIT",
5    "scripts": {
6      "ng": "ng",
7      "start": "ng serve",
8      "build": "ng build",
9      "test": "ng test",
10     "lint": "ng lint",
11     "e2e": "ng e2e"
12   },
13   "private": true,
14   "dependencies": {
15     "@angular/animations": "^4.0.0",
16     "@angular/common": "^4.0.0",
17     "@angular/compiler": "^4.0.0",
18     "@angular/core": "^4.0.0",
19     "@angular/forms": "^4.0.0",
20     "@angular/http": "^4.0.0",
21     "@angular/platform-browser": "^4.0.0",
22     "@angular/platform-browser-dynamic": "^4.0.0",
23     "@angular/router": "^4.0.0",
24     "core-js": "^2.4.1",
25     "rxjs": "^5.4.1",
26     "zone.js": "^0.8.14",
27     "bootstrap": "3.3.7",
28     "@types/bootstrap": "3.3.36"
29   },
30   "devDependencies": {
31     "@angular/cli": "1.2.6",
32     "@angular/compiler-cli": "^4.0.0",
33     "@angular/language-service": "^4.0.0",
```

Structuur onder de 'src' map

- **Index.html:**
 - De ENIGE html pagina die wordt ingeladen
 - Moet niet worden aangepast door de ontwikkelaar
 - Alle html, css, javascript wordt geïnjecteerd door het angular framework 'at runtime'
- **Main.ts:**
 - Het typescript bestand dat wordt uitgevoerd bij het opstarten.
 - Moet niet worden aangepast door de ontwikkelaar
- **Styles.css:**
 - De globale style sheet, hier kunnen styles worden toegevoegd die geldig zijn voor de hele toepassing
- **App** map: hieronder plaatsen we alle modules en componenten van onze toepassing:
 - Typescript bestanden, html, css,..



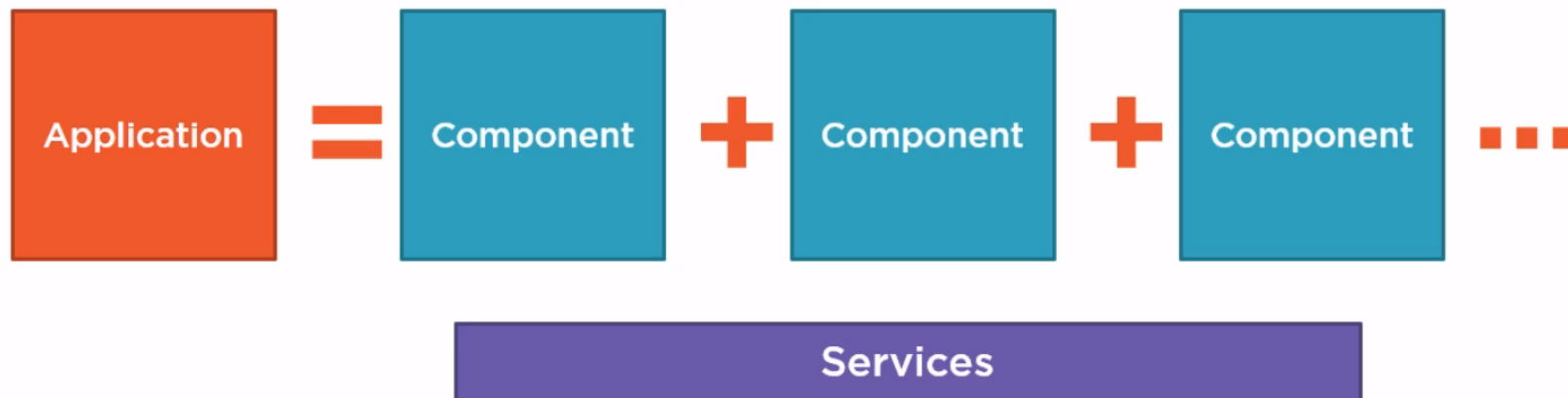
Inhoud

1. Deel 1

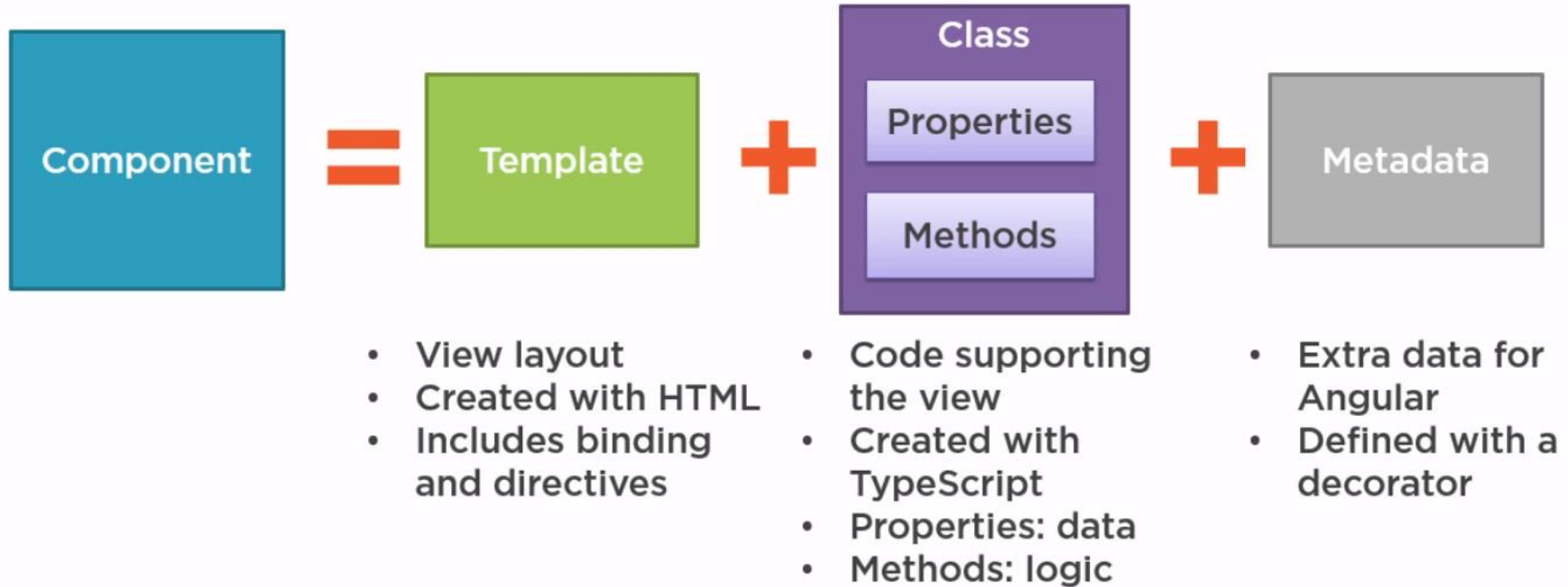
1. Waarom Angular ?
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - **Van een Applicatie**
 - **Van een Component**
4. Nieuwe Component Aanmaken

3.1. Hoe is een applicatie opgebouwd ?

Anatomy of an Angular Application



3.2. Hoe is een Angular component opgebouwd ?



Voorbeeld van een Component

TS app.component.ts ✕

<> app.component.html

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'My first application with Angular';
10 }
11
```

Import

Metadata
& Template

Class

Merk op dat de (app.component) html en css bestanden uiteraard ook nog deel uitmaken van deze 'Component', alhoewel deze bestanden hier niet expliciet worden getoond.

Inhoud

1. Deel 1

1. Waarom Angular ?
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - Van een Applicatie
 - Van een Component
4. **Nieuwe Component Aanmaken**

4.1. Nieuwe component aanmaken

- Je kan best elke oefening in een andere component aanmaken, op die manier oefen je tevens ook het aanmaken van nieuwe componenten.
- Via angular-cli
 - **ng generate component [Naam]** (of korte versie: **ng g c [naam]**)
 - Er wordt een submapje aangemaakt voor elke component: **src/app/[Naam]**
 - Via de 'selector' (die je desgevallend nog kan hernoemen) kan je vervolgens de component ergens gaan weergeven in de toepassing. De selector is eigenlijk een nieuw HTML element.

```
@Component({  
  selector: 'app-second',  
  templateUrl: './second.component.html',  
  styleUrls: ['./second.component.css']  
})  
export class SecondComponent {
```

```
<div>  
  <app-second></app-second>  
</div>
```

