

# **Web Frameworks Les 1**

## **Fundamenten van moderne webontwikkeling**

*Kristof Michiels (team: Sven Mariën, Andie Similon)*

# Topics

- Van **client-server paradigma** naar **frameworks**
- Evolutie van het web → libraries → grote frameworks
- MEAN-stack & Single-Page Applications (SPA) als architectuurpatronen
- **TypeScript & transpiling** als basis voor robuuste code
- Uitdagingen & kansen in het **AI-tijdperk**
- Aanpak binnen het OLOD

# Fundamente(n) van moderne webontwikkeling

# Web Frameworks?

- Doel: **leren werken met een modern front-end framework** (Angular)
- Bouwt verder op:
  - **Web Technology** → HTML, CSS, DOM scripting
  - **Web Programming** → JavaScript/TypeScript, Node, Express
- Angular gebruikt het web als **volwaardige applicatie-omgeving**
- Werkt binnen het **Node-ecosysteem** (npm, tooling)
- Past in het klassieke **client-server paradigma**

# Client–Server in web-context

- **Client (browser)**
  - Toont de UI aan de gebruiker
  - Vraagt data of acties op bij de server
  - Draait Angular-applicatie (SPA)
- **Server (Node, API, database, ...)**
  - Verwerkt aanvragen van de client
  - Levert data terug via **HTTP / REST / GraphQL**
  - Kan logica & beveiliging afhandelen

# Client–Server in web-context



- **Angular's rol**
  - **Frontend**: draait volledig in de client
  - Verbindt met back-end via **API-calls**
  - Maakt van de browser een **echte applicatieomgeving**

# Wat zijn web frameworks?

- **Evolutie van het web**

- Statisch (HTML) → dynamisch (JavaScript) → gestructureerd (frameworks)
- Libraries: **jQuery** → eenvoudiger DOM-manipulatie
- Moderne libraries: **React** → declaratieve UI
- Frameworks: **Angular** → volledige architectuur

# Wat zijn web frameworks?

- **Waarom frameworks?**
  - Bieden **structuur & schaalbaarheid**
  - Bevorderen **hergebruik & onderhoudbaarheid**
  - Ondersteunen het bouwen van **volwaardige applicaties** in de browser



# Wat zijn web frameworks?

- **De grote spelers vandaag**
  - **Angular** → compleet ecosysteem, enterprise-grade
  - **React** → flexibel, bibliotheken naar keuze
  - **Vue** → lichtgewicht, lage instapdrempel
- **Keuzes & kritiek**
  - Functioneel kunnen ze vaak **hetzelfde**
  - Verschil zit in **aanpak, tooling en community**
  - **Trade-off**: vrijheid (React/Vue) ↔ structuur (Angular)

# Wat zijn web frameworks?

- **Toekomst**
  - **Meta-frameworks:** Next.js, Nuxt, Angular Universal
  - Focus op **snellere setup, automatische optimalisatie,**  
en **AI-assist bij development**

# De MEAN-stack 🥘

- **MongoDB** → NoSQL database
  - **Express.js** → back-end web framework (Nest.js = moderne implementatie)
  - **Angular** → front-end framework
  - **Node.js** → runtime voor server-side JavaScript
- 👉 Samen vormen ze een **volledige stack**: van database tot UI

# Single-Page Applications (SPA)



- **Één HTML-pagina** → dynamisch bijgewerkt in de browser
- Navigatie & content = client-side geregeld
- **Voordelen:** snelle UX, minder server load, app-gevoel
- **Uitdagingen:** SEO, initial load, security
  - 👉 Angular = framework bij uitstek voor **SPA's**

# TypeScript

- Superset van JavaScript met **types**
- Voordelen:
  - **Type-safety** → fouten sneller opsporen
  - **Betere tooling** → IntelliSense, autocompletion
  - **Leesbaarheid & onderhoudbaarheid**
- Angular = volledig gebouwd in TypeScript

👉 Voor developers: **veiliger en productiever coderen**

# Transpiling met TypeScript

- Browsers begrijpen enkel **JavaScript**
  - TypeScript code → **transpiled** naar JavaScript
  - Tooling: **tsc** (TypeScript compiler) of via Angular CLI
  - Debuggen in TypeScript dankzij **source maps**
- 👉 Transpiling = brug tussen moderne code en brede browser-compatibiliteit

# **Uitdagingen & Aanpak**

# IT in 2025 – Uitdagingen

- **Frameworks blijven cruciaal**
  - Basis voor moderne, schaalbare apps
  - Evolueren mee met AI & meta-frameworks
- **AI-tijdperk**
  - GenAI kan code genereren → maar *jij* moet de structuur begrijpen
  - Frameworkkennis = **basisvaardigheid** voor elke IT'er



# IT in 2025 – Uitdagingen

- **Uitdaging voor studenten**
  - **Leercurve Angular:** stevig → maar haalbaar met discipline
  - **Head-first aanpak:** leren door doen, projectmatig werken
  - **Focus & aanwezigheid** zijn sleutel → niet meedoen = niet slagen
  - Tijdsinvestering loont: eenmaal de basis gekend → productiviteit  $\times 10$

# Front-end is Powerful ⚡

- **Waar developer en gebruiker elkaar ontmoeten**
- Onmisbare skill voor **élke IT'er**
  - Interfaces bouwen die data & logica toegankelijk maken
  - Essentieel in AI-, IoT- en Cyber Security-apps
- Deze cursus = jouw eerste stap naar:
  - Zelf Angular-apps ontwikkelen
  - Comfortabel werken met moderne frameworks

👉 **Front-end = impact maken als developer**

# Angular ook voor AI, Cyber Sec & IoT

- **Moderne, performante apps** → basis in elk IT-domein
  - **IoT** → dashboards, real-time monitoring & device control
  - **Cyber Security** → integratie van authenticatie, encryptie & veilige flows
  - **AI** → frontend voor chatbots, dashboards, data-analyse
- 👉 Angular = de **schakel** tussen technologie & gebruiker

# Onze aanpak

- **Duiding** → heldere uitleg van kernconcepten
  - **Voorbeelden** → kleine demo-apps die de theorie concreet maken
  - **Labo's** → hands-on opdrachten om zelf te oefenen
  - **Groei** → van eenvoudige componenten naar eigen toepassingen
- 👉 Einddoel: jij kan zelfstandig een Angular-app bouwen