

# Labo 2 – Angular basis verdieping

---

In dit labo maak je drie toepassingen (+ 1 vrijblijvende) die elk een belangrijk Angular-concept inoefenen. Je werkt telkens met een kleine dataset en meerdere componenten.

Na dit labo kan je:

- Een app opdelen in meerdere componenten en data via `@Input` doorgeven.
- Bindingen gebruiken (`two-way`, `event`, `@if`, `@switch`).
- Routes configureren en URL-parameters gebruiken.

## Opdracht 1 – Receptenapp

**Doel:** werken met parent-child componenten, `@Input`, `@for`, `@if` en een herbruikbaar component.

### Componenten

- `AppComponent` (root)
- `ReceptenLijstComponent` (toont de lijst met recepten)
- `ReceptKaartComponent` (child: details van één recept)
- `LabelComponent` (herbruikbaar: badges voor labels zoals "Vegetarisch", "Snel klaar")

### Dataset

```
recepten = [  
  { id: 1, titel: 'Spaghetti Bolognese', tijd: 30, labels: ['Italiaans', 'Snel'] },  
  { id: 2, titel: 'Groene Curry', tijd: 25, labels: ['Thais', 'Pittig'] },  
  { id: 3, titel: 'Chili sin carne', tijd: 40, labels: ['Vegetarisch'] },  
  { id: 4, titel: 'Lasagne', tijd: 60, labels: ['Italiaans'] },  
  { id: 5, titel: 'Caesarsalade', tijd: 15, labels: ['Snel', 'Gezond'] }  
];
```

### Taken

- Toon alle recepten met `@for`.
- Geef titel en kooktijd via `@Input` door aan `ReceptKaartComponent`.
- Gebruik `LabelComponent` om de labels weer te geven.
- Als er geen recepten zijn, toon een boodschap met `@if`.
- Styling is van kleiner belang maar het is toegelaten je applicatie netjes te stylen 😊

## Opdracht 2 – Kleine Quiz-toepassing

**Doel:** oefenen met two-way binding, event binding en control flow (`@if`, `@switch`).

### Component

- **QuizComponent** (alle logica en template in één component).

## Dataset

Stel zelf een aantal leuke vragen op! De vragen hier gelden enkel als voorbeeld.

```
vragen = [  
  { id: 1, vraag: 'Wat is 2 + 2?', antwoord: '4' },  
  { id: 2, vraag: 'Hoofdstad van België?', antwoord: 'Brussel' },  
  { id: 3, vraag: 'Welke kleur krijg je door blauw + geel te mengen?',  
    antwoord: 'Groen' },  
  { id: 4, vraag: 'Wat is 10 gedeeld door 2?', antwoord: '5' }  
];
```

## Taken

- Toon telkens één vraag (`vragen[huidige]`).
- Inputveld met `[(ngModel)]` voor het antwoord.
- Knop (`click`) om het antwoord te controleren.
- Toon feedback met `@if`.
- Gebruik `@switch`: zolang er vragen zijn → toon de vraag, anders → toon de eindscore.
- Styling is van kleiner belang maar het is toegelaten je applicatie netjes te stylen 😊

## Opdracht 3 – Gebruikersapp met routing

**Doel:** routing, `routerLink`, `router-outlet`, routeparameters.

## Componenten

- **StartComponent** → toont een overzichtslijst van gebruikers.
- **GebruikerDetailComponent** → detailpagina van één gebruiker, via URL-parameter.
- **NietGevondenComponent** → foutpagina.

## Routes

```
export const routes: Routes = [  
  { path: '', component: StartComponent },  
  { path: 'gebruiker/:id', component: GebruikerDetailComponent },  
  { path: '**', component: NietGevondenComponent },  
];
```

Dataset (in beide componenten hardcoded)

```
gebruikers = [  
  { id: 1, naam: 'Alice Janssens', email: 'alice@example.com' },  
  { id: 2, naam: 'Bob Peeters', email: 'bob@example.com' },  
];
```

```
{ id: 3, naam: 'Charlie Claes', email: 'charlie@example.com' },
{ id: 4, naam: 'Dorien Jacobs', email: 'dorien@example.com' },
{ id: 5, naam: 'Elias Vermeulen', email: 'elias@example.com' },
{ id: 6, naam: 'Fatima Ali', email: 'fatima@example.com' },
{ id: 7, naam: 'Gert Vandenberghe', email: 'gert@example.com' },
{ id: 8, naam: 'Hannah Willems', email: 'hannah@example.com' },
{ id: 9, naam: 'Ibrahim Mohamed', email: 'ibrahim@example.com' },
{ id: 10, naam: 'Julie Maes', email: 'julie@example.com' }
];
```

## Taken

- **StartComponent**: toon alle gebruikers met `@for`, met links naar `/gebruiker/<id>`.
- **GebruikerDetailComponent**: lees `id` uit via `ActivatedRoute` en toon de juiste gebruiker.
- **NietGevondenComponent**: eenvoudige melding.

## Vrijblijvend - opdracht 4 – Boekenapp met filter

**Doel**: combineren van componenten, `@Input`, `@for`, `@if`, two-way binding en event binding.

## Componenten

- **BoekenLijstComponent** → toont alle boeken.
- **BoekKaartComponent** → child-component met details van één boek.
- **FilterComponent** → invoerveld waarmee een zoekterm wordt ingegeven.

## Dataset

```
boeken = [
  { id: 1, titel: 'De Avonden', auteur: 'Gerard Reve', genre: 'Roman' },
  { id: 2, titel: 'Het Diner', auteur: 'Herman Koch', genre: 'Thriller' },
  { id: 3, titel: 'Oorlog en Vrede', auteur: 'Leo Tolstoj', genre:
'Klassieker' },
  { id: 4, titel: 'De kleine prins', auteur: 'Antoine de Saint-Exupéry',
genre: 'Kinderboek' },
  { id: 5, titel: 'Harry Potter en de Steen der Wijzen', auteur: 'J.K.
Rowling', genre: 'Fantasy' },
  { id: 6, titel: 'Nooit meer slapen', auteur: 'W.F. Hermans', genre:
'Roman' },
  { id: 7, titel: 'De Hobbit', auteur: 'J.R.R. Tolkien', genre: 'Fantasy'
},
  { id: 8, titel: 'Max Havelaar', auteur: 'Multatuli', genre: 'Roman' }
];
```

## Taken

- **BoekenLijstComponent**
  - Toon alle boeken met `@for`.

- Gebruik **BoekKaartComponent** om titel, auteur en genre te tonen via **@Input**.
- Als er geen resultaten zijn, toon een melding met **@if**.

- **FilterComponent**

- Bevat een inputveld met **[(ngModel)]="zoekterm"**.
- Via **(input)** event wordt de zoekterm doorgegeven aan de parent.

- **Extra logica**

- Pas in **BoekenLijstComponent** een eenvoudige filter toe: toon enkel boeken waarvan de titel of auteur de zoekterm bevat.
- De lijst moet live updaten bij het typen.