

**ilionx**



Global Knowledge®

# Angular Advanced Short recap day #1

Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

WORLDWIDE LOCATIONS

BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR  
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

# Day 1: Evaluation

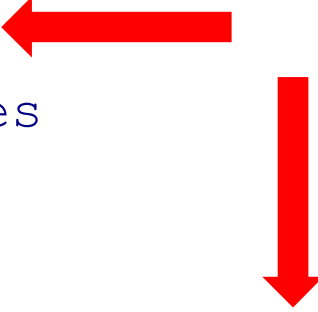
- *Vanuit Ilionx: Carin, Jorik, Fer, Stefan, Tawwab*
- Aandachtspunten:
  1. Betere **interactie** met aanwezigen
    - Wil je je camera aanzetten? (beter dan een box met je naam, meer persoonlijk gevoel bij de deelnemers)
    - Maak gebruik van de chatbox en microfoon
    - Recording om later terug te kijken – nodig?
  2. **Breakout rooms** – Overleggen in kleine groepjes, behoefte aan?
  3. **Alternatieve data**? 20-21 april is ver weg. Overige opties:
    - 7 april (dinsdag)
    - 8 april (woensdag)
    - 9 april (donderdag)
    - 15 april (woensdag)
    - 16 april (donderdag)

# Recap day 1: **Architecture**

- NG applications with multiple modules
  - Using multiple modules in your app
  - Modules and the router
- Lazy loading modules
  - `PreloadAllModules` & Custom Loading strategies
- Content Projection
  - `<ng-content select="...">`

# Configuring the router

```
let routerConfig: ExtraOptions = {  
  enableTracing: true  
  preloadingStrategy: PreloadAllModules  
};  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes, routerConfig)],  
  exports: [RouterModule]  
})
```

Two red arrows originate from the right side of the code. One arrow points horizontally to the left, ending at the 'routerConfig' argument in the 'RouterModule.forRoot' call. The other arrow points vertically downwards, ending at the 'RouterModule' argument in the same call.

Routing events are actually **Observables**.

Which means we can subscribe! And do something like:

```
this.router.events
    .subscribe(event =>{
        console.log(' router event: ', event);
    })
```

(Don't forget to inject router:

```
constructor(private router: Router) {  })
```

Or, in a more reactive way of programming:

```
this.router.events.pipe(  
  filter(event => event instanceof NavigationEnd),  
  map(...),  
  ...  
) .subscribe(event => {  
  console.log(' router event: ', event  
});
```

Handle events as desired - logging, setting variables, updating state, ...

# Agenda - 4 nights - Thematic



- Day 1: Architecture
  - Composing Applications with multiple modules
  - Routing and lazy loading modules
  - Loading Strategies
  - Advanced components
- Day 2: **Observables**
  - Observables from scratch, more operators
  - Examples (typeahead, fetching data from multiple sources)

# Wrap up

- Creating observables from scratch
- Working with streams
- Operators
  - `map()`, `mapTo()`
  - `filter()`
  - `concat()`
  - `scan()`
  - `merge()`, `mergeMap()`, `switchMap()`, ...
- Working with http-requests



# Next up

- Day 3: **State Management**
  - State management with `@ngrx/store`
  - Concepts – actions, dispatchers, reducers, store
  - Payload, complex types
- Day 4: **Enterprise applications**
  - Working with monorepo's, Angular CLI
  - More on Angular Schematics

**ilionx**



Global Knowledge®

# See you then!

Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

WORLDWIDE LOCATIONS

BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR  
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA