meat & more

Global Knowledge®

Angular Advanced
Short recap day #1

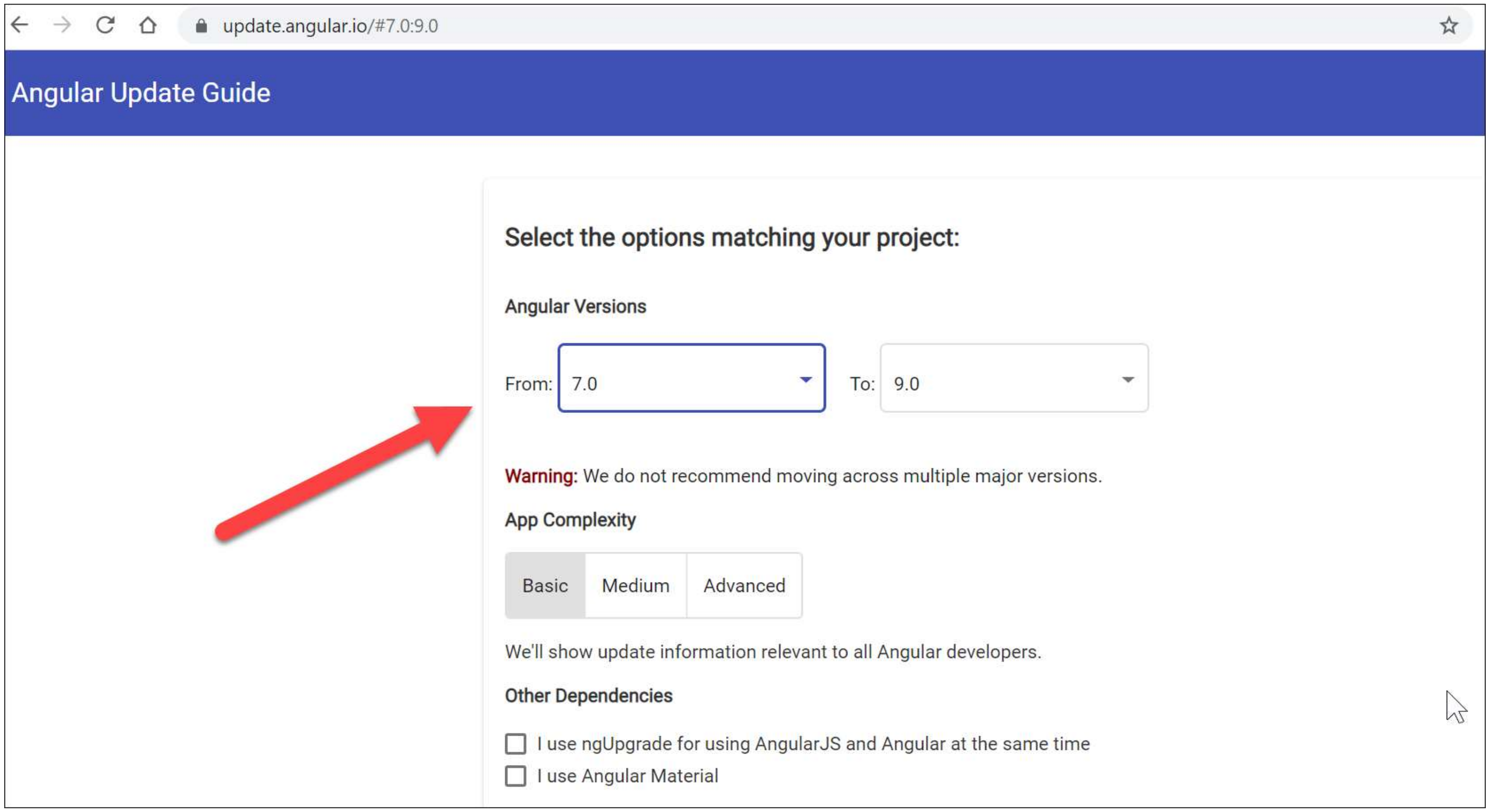
Peter Kassenaar – info@kassenaar.com

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="…">`
- Smart/View components
  - Smart: mostly logic, little UI
  - View: mostly UI, little logic

On `ng update`

- DON'T try to update over major versions (e.g. from NG 7 → NG 9)


update.angular.io/#7.0:9.0

Angular Update Guide

Select the options matching your project:

Angular Versions

From: 7.0 To: 9.0

Warning: We do not recommend moving across multiple major versions.

App Complexity

[✓ Basic] Medium Advanced

We'll show update information relevant to all Angular developers.

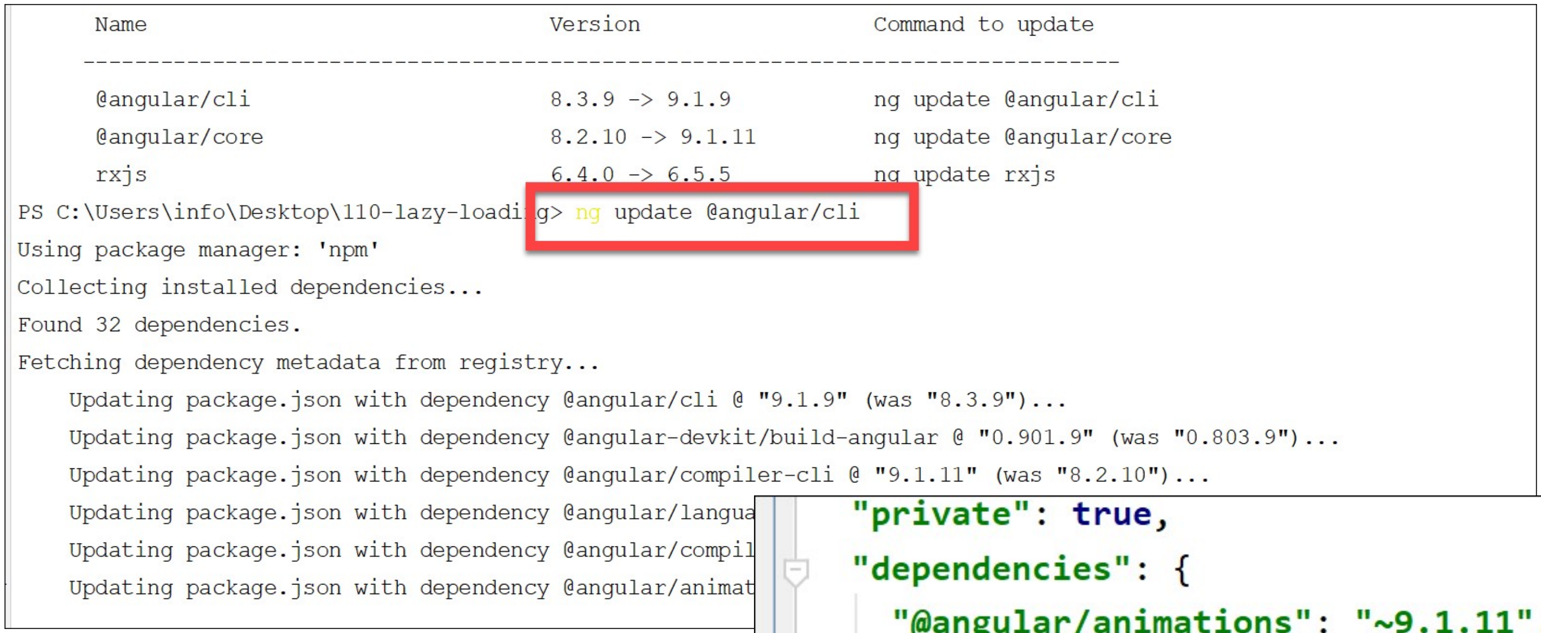
Other Dependencies

☐ I use ngUpgrade for using AngularJS and Angular at the same time

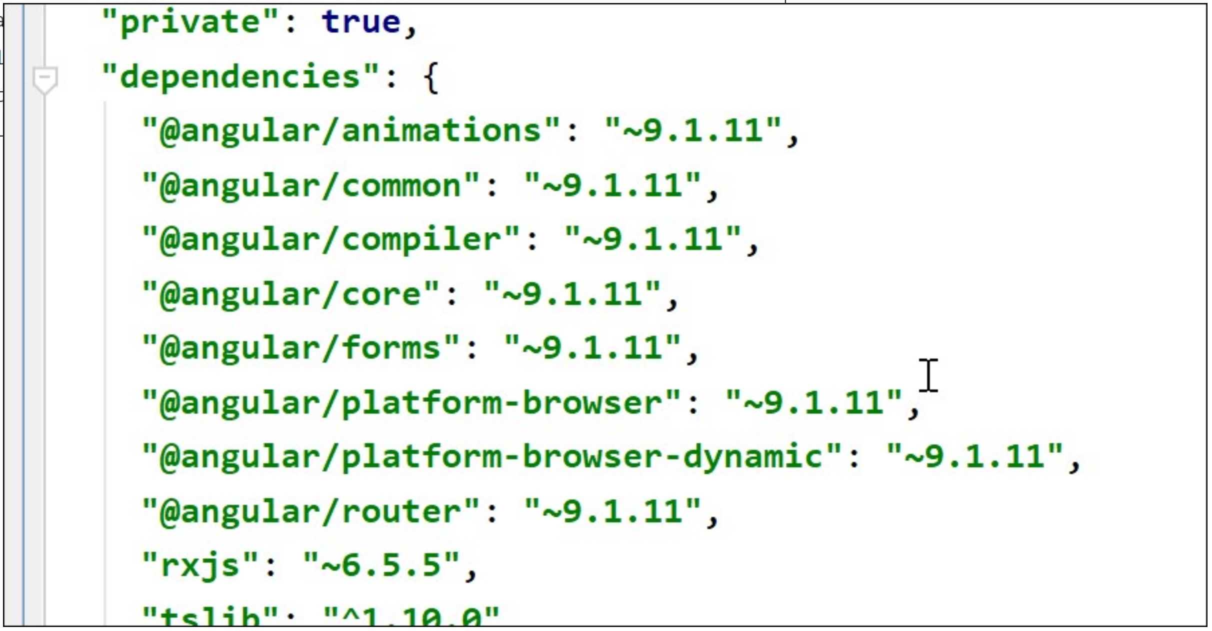
☐ I use Angular Material

https://update.angular.io/

`*ngIf` not working anymore after update?

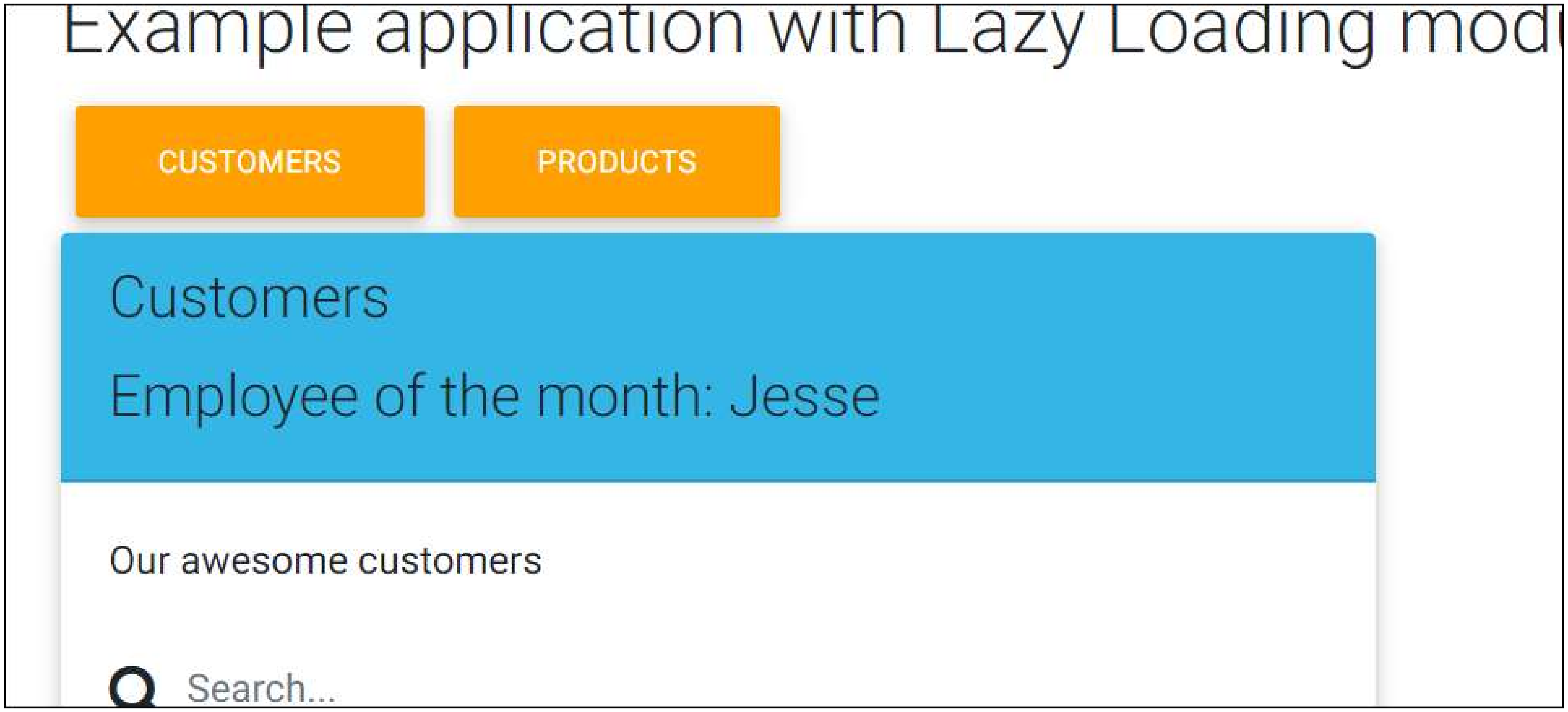


```
      Name                                  Version                    Command to update
     --------------------------------------------------------------------------------
      @angular/cli                          8.3.9 -> 9.1.9             ng update @angular/cli
      @angular/core                         8.2.10 -> 9.1.11           ng update @angular/core
      rxjs                                  6.4.0 -> 6.5.5             ng update rxjs
PS C:\Users\info\Desktop\110-lazy-loading> ng update @angular/cli
Using package manager: 'npm'
Collecting installed dependencies...
Found 32 dependencies.
Fetching dependency metadata from registry...
    Updating package.json with dependency @angular/cli @ "9.1.9" (was "8.3.9")...
    Updating package.json with dependency @angular-devkit/build-angular @ "0.901.9" (was "0.803.9")...
    Updating package.json with dependency @angular/compiler-cli @ "9.1.11" (was "8.2.10")...
    Updating package.json with dependency @angular/langua
    Updating package.json with dependency @angular/compil
    Updating package.json with dependency @angular/animat
```



```
  "private": true,
  "dependencies": {
    "@angular/animations": "~9.1.11",
    "@angular/common": "~9.1.11",
    "@angular/compiler": "~9.1.11",
    "@angular/core": "~9.1.11",
    "@angular/forms": "~9.1.11",
    "@angular/platform-browser": "~9.1.11",
    "@angular/platform-browser-dynamic": "~9.1.11",
    "@angular/router": "~9.1.11",
    "rxjs": "~6.5.5",
    "tslib": "^1.10.0"
```

Verdict – sorry, not reproducable… (?)



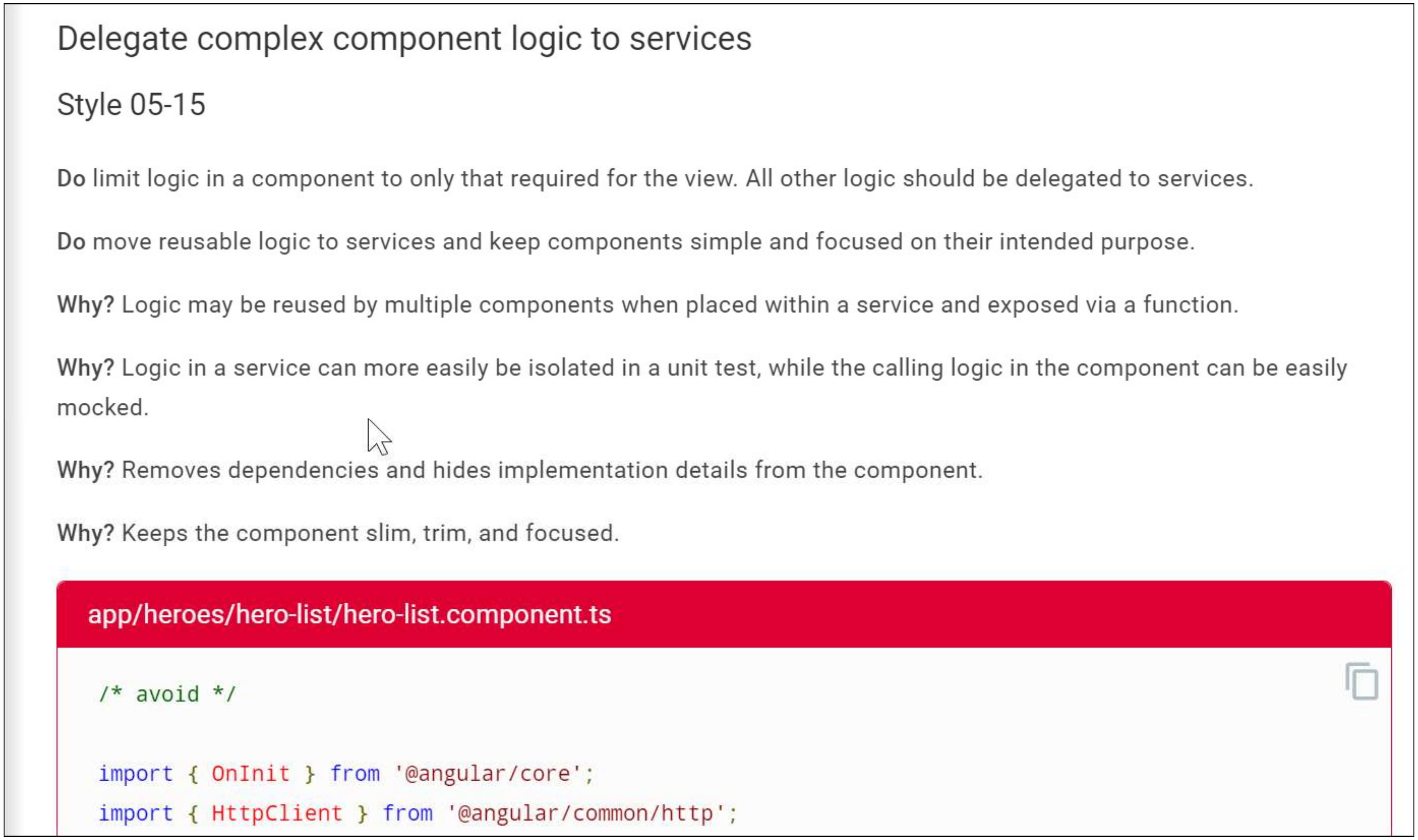
Example application with Lazy Loading mod

Customers

Employee of the month: Jesse

Our awesome customers

Logic in Services or Components? Style Guide like…



Delegate complex component logic to services

Style 05-15

Do limit logic in a component to only that required for the view. All other logic should be delegated to services.

Do move reusable logic to services and keep components simple and focused on their intended purpose.

Why? Logic may be reused by multiple components when placed within a service and exposed via a function.

Why? Logic in a service can more easily be isolated in a unit test, while the calling logic in the component can be easily mocked.

Why? Removes dependencies and hides implementation details from the component.

Why? Keeps the component slim, trim, and focused.

app/heroes/hero-list/hero-list.component.ts

```
/* avoid */

import { OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

https://angular.io/guide/styleguide#delegate-complex-component-logic-to-services

Agenda - 3 days - Thematic

THEMA

- Day 1: Architecture
  - Composing Applications with multiple modules
  - Routing and lazy loading modules
  - Loading Strategies
  - Advanced components
- Day 2: Observables
  - What is the observable pattern?
  - Observables from scratch, RxJS-operators
  - Examples (typeahead, fetching data from multiple sources)
  - Time permitting – Enterprise/monorepo applications

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: Miscellaneous
  - Working with 3rd party libraries
  - Unit Testing w/ Karma & Jasmine, Cypress as alternative
  - Q & A
  - Introduction - @ngrx/store
  - More on Angular Schematics
  - …