

```
setOnLongClickListener(this)
```

6 Tips pour améliorer

sa web app Angular



takima

Camille Pillot

Développeuse web

✉ cpillot@takima.fr

🐦 [@CamiPillotDev](https://twitter.com/CamiPillotDev)

in <https://www.linkedin.com/in/camillepillot/>





takima

Camille Pillot

Développeuse web



cpillot@takima.fr



@CamiPillotDev



6 tips avec Angular

Pour toi





Les Alias

Les alias

Mes imports avant

```
import { Employee } from '../../../../../core/models/employee';
import { EmployeeService } from '../../../../../core/service/employee.service';
import { Banner } from '../../../../../shared/banner';
import { Mood } from '../models/mood';
import { MoodService } from '../mood.service';
```



tsconfig.ts

Les alias

Mon tsconfig

```
"compilerOptions": {  
    "outDir": "./dist/out-tsc",  
    "strict": true,  
    "target": "es2017",  
    "module": "es2020",  
    "lib": [ "es2020", "dom" ]  
},
```

Les alias

Mon tsconfig

```
"compilerOptions": {  
    "baseUrl": ".",
    "paths": {
        "@app/*": ["src/app/*"],
        "@core/*": ["src/core/*"],
        "@shared/*": ["src/shared/*"],
    },
    "outDir": "./dist/out-tsc",
    "strict": true,
    "target": "es2017",
    "module": "es2020",
    "lib": [ "es2020", "dom" ]
},
```

Les alias

Mes imports avant

```
import { Employee } from '../../core/models/employee';
import { EmployeeService } from '../../core/service/employee.service';
import { Banner } from '../../shared/banner';
import { Mood } from '../models/mood';
import { MoodService } from '../mood.service';
```

Les alias

Mes imports après

```
import { Employee } from '../../core/models/employee';
import { EmployeeService } from '../../core/service/employee.service';
import { Banner } from '../../shared/banner';
import { Mood } from '../models/mood';
import { MoodService } from '../mood.service';
```



```
import { Employee } from '@core/models/employee';
import { EmployeeService } from '@core/service/employee.service';
import { Banner } from '@shared/models/banner';
import { Mood } from '@app/models/mood';
import { MoodService } from '@app/mood.service';
```

Les alias

Mes imports après

```
import { Employee } from '../../core/models/employee';
import { EmployeeService } from '../../core/service/employee.service';
import { Banner } from '../../shared/banner';
import { Mood } from '../models/mood';
import { MoodService } from '../mood.service';
```



```
import { Employee } from '@core/models/employee';
import { EmployeeService } from '@core/service/employee.service';
import { Banner } from '@shared/models/banner';
import { Mood } from '@app/models/mood';
import { MoodService } from '@app/mood.service';
```



Mes imports de style avant

```
@import '../../../../../styles/special-theme/colors.scss';
@import '../../../../../styles/variables/variable.scss';

.banner {
    color: $text-default-color;
    padding: $large-padding;
}
```

Les alias

Solution



angular.json

Les alias

Mon angular.json

```
"projects": {  
  "alias": {  
    ...  
    "architect": {  
      "build": {  
        "options": {  
          ...  
          "styles": [  
            "src/styles.scss"  
          ],  
          "scripts": []  
        },  
        ...  
      }  
    }  
  }  
}
```

Les alias

Mon angular.json

```
"projects": {  
  "alias": {  
    ...  
    "architect": {  
      "build": {  
        "options": {  
          ...  
          "styles": [  
            "src/styles.scss"  
          ],  
          "stylePreprocessorOptions": {  
            "includePaths": [  
              "src/styles/special-theme",  
              "src/styles/variables"  
            ]  
          },  
          "scripts": []  
        },  
        ...  
      }  
    }  
  }  
}
```

Mes imports de style avant

```
@import '../../styles/special-theme/colors.scss' ;  
@import '../../styles/variables/variable.scss' ;
```

Les alias

Mes imports de style après

```
@import '../../../../../styles/special-theme/colors.scss' ;  
@import '../../../../../styles/variables/variable.scss' ;
```



```
@import 'colors';  
@import 'variables';
```

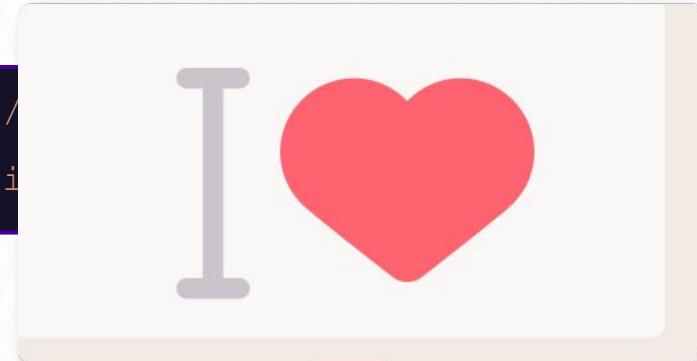
Les alias

Mes imports de style après

```
@import '../../../../../styles/special-theme/  
@import '../../../../../styles/variables/vari
```



```
@import 'colors';  
@import 'variables';
```





TrackBy

Démo pour mettre en vert ce qui est re-render

The screenshot shows a web browser window with the URL `localhost:4200`. The page title is "Management de mes étudiants". Below the title, there is a list of five students:

- Alain Térieur, alainterior@gmail.com, with a "Détails" button.
- Alex Térieur, alexterior@gmail.com, with a "Détails" button and a small crown icon.
- Cara Melmou, caramelmou@gmail.com, with a "Détails" button.
- Jean Peupu, jeanpeupu@gmail.com, with a "Détails" button.
- Lara Tatouille, laratatouille@gmail.com, with a "Détails" button.

Below the student list is a form titled "Sélectionner l'étudiant du mois :". It contains a list of radio buttons for the students: Alain, Alex (selected), Cara, Jean, and Lara.

On the right side of the browser window, the DevTools "More tools" panel is open. The "Elements" tab is selected, showing the DOM structure:

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <app-root _nghost-hvc-c69 ng-version="13.3.12">
      <shell _ngcontent-hvc-c69 _ngcontent-hvc-c68>
        <div> ...
    </body>

```

The "More tools" panel also includes sections for "Paint flashing", "Layout Shift Regions", "Layer borders", "Frame Rendering Stats", "Scrolling performance issues", and "Highlight ad frames". A yellow brush stroke is visible on the bottom right corner of the slide.

Démo sans trackBy

localhost:4200

TakiPortal Angular conf Advocacy

Management de mes étudiants

Alain Térieur alainterier@gmail.com	Détails	Alex Térieur alexterieur@gmail.com	Détails
Cara Melmou caramel mou@gmail.com	Détails	Jean Peupu jeanpeupu@gmail.com	Détails
Lara Tatouille laratatouille@gmail.com	Détails		

Sélectionner l'étudiant du mois :

- Alain
- Alex
- Cara
- Jean
- Lara

0:27 / 1:16

Elements Console ↻ 1 ⚙

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
    <app-root _nghost-hvc-c69 ng-version="13.3.12">
      ...
    </app-root>
  </head>
  <body>
    <div _ngcontent-hvc-c69>
      ...
    </div>
  </body>
</html>
```

Styles Computed Layout Event Listeners

Filter :hover .cls +

```
.ghost-hvc-c681 {
  display: flex;
  flex-direction: column;
  width: 100%;
  height: 100%;
}
```

Inherited from app-root

```
.ghost-hvc-c691 {
  ...
}
```

Console Rendering Animations Issues

Paint flashing Highlights areas of the page (green) that need to be repainted. May not be suitable for people prone to photosensitive epilepsy.

Layout Shift Regions Highlights areas of the page (blue) that were shifted. May not be suitable for people prone to photosensitive epilepsy.

Layer borders Shows layer borders (orange/olive) and tiles (cyan).

Frame Rendering Stats Plots frame throughput, dropped frames distribution, and GPU memory.

Scrolling performance issues Highlights elements (teal) that can slow down scrolling, including touch & wheel event handlers and other main-thread scrolling situations.

Highlight ad frames Highlights frames (red) detected to be ads.

Core Web Vitals Shows an overlay with Core Web Vitals.

HD

TrackBy

Démo ajout du trackBy dans le html

The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER:** Shows the project structure with files like `ALIAS`, `src`, `app`, `details`, `home`, and `assets`.
- home.component.html:** Contains the following HTML code:

```
<div class="student-page">
  <div *ngFor="let student of students; trackBy: trackByFn" class="student-list">
    <div class="student-identity">
      <span>{{ student.firstname }} {{ student.lastname }}</span>
      <span>{{ student.email }}</span>
    </div>

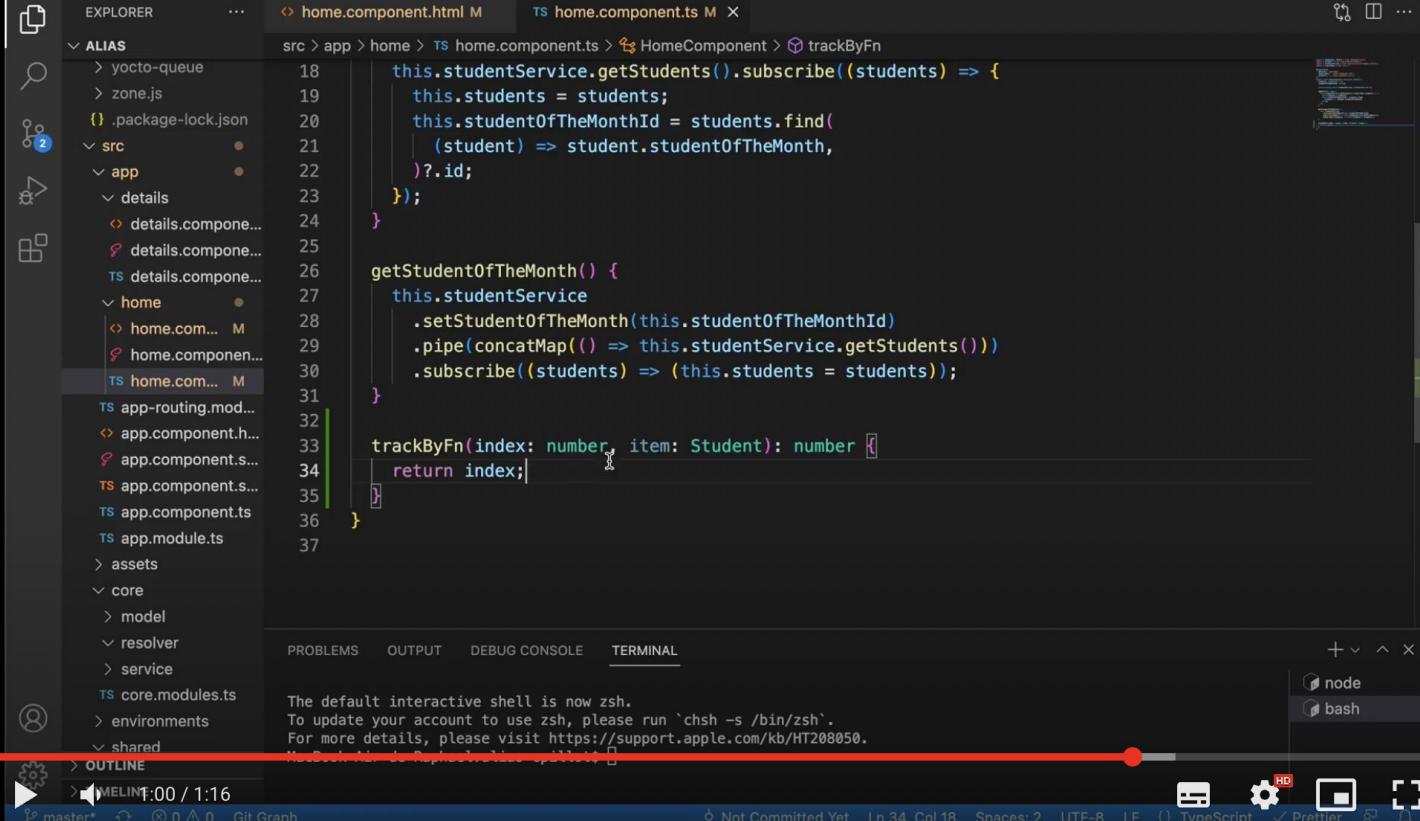
    <a class="button" [routerLink]="['student', student.id]">Détails</a>
  </div>
</div>

<fieldset class="fieldset">
  <legend>Sélectionner l'étudiant du mois :</legend>
  <div *ngFor="let student of students">
    <input
      type="radio"
      [id]="student.id"
    </input>
  </div>
</fieldset>
```
- home.component.ts:** Contains the following TypeScript code:

```
1  <div class="student-page">
2    <div *ngFor="let student of students; trackBy: trackByFn" class="student-list">
3      <div class="student-identity">
4        <span>{{ student.firstname }} {{ student.lastname }}</span>
5        <span>{{ student.email }}</span>
6      </div>
7
8      
13
14      <a class="button" [routerLink]="['student', student.id]">Détails</a>
15    </div>
16  </div>
17
18  <fieldset class="fieldset">
19    <legend>Sélectionner l'étudiant du mois :</legend>
20    <div *ngFor="let student of students">
21      <input
22        type="radio"
23        [id]="student.id"
24      </input>
25    </div>
26  </fieldset>
```
- TERMINAL:** Displays the message: "The default interactive shell is now zsh. To update your account to use zsh, please run `chsh -s /bin/zsh`. For more details, please visit https://support.apple.com/kb/HT208050".
- Bottom Status Bar:** Shows the file path as `MacBook-Air-25:React-App camillepillot`, the status `0:43 / 1:16`, and icons for Git, Prettier, and other development tools.

TrackBy

Démo ajout de la fonction trackByFn dans le ts



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure with files like `home.component.html`, `home.component.ts`, `app-routing.module.ts`, etc.
- home.component.ts** editor tab: Displays the TypeScript code for the component.

```
src > app > home > TS home.component.ts M
18  this.studentService.getStudents().subscribe(students) => {
19    this.students = students;
20    this.studentOfTheMonthId = students.find(
21      student) => student.studentOfTheMonth,
22      )?.id;
23    );
24  }

25
26  getStudentOfTheMonth() {
27    this.studentService
28      .setStudentOfTheMonth(this.studentOfTheMonthId)
29      .pipe(concatMap(() => this.studentService.getStudents()))
30      .subscribe(students) => (this.students = students));
31  }

32
33  trackByFn(index: number, item: Student): number {
34    return index;
35  }
36
37 }
```
- TERMINAL**: Shows the message: "The default interactive shell is now zsh. To update your account to use zsh, please run `chsh -s /bin/zsh`. For more details, please visit https://support.apple.com/kb/HT208050".
- Bottom Status Bar**: Shows the file is not committed yet, line 34, column 18, spaces 2, LF, TypeScript, Prettier, and other status icons.

TrackBy

Démo avec trackBy

The screenshot shows a web application running at `localhost:4200`. The title bar indicates the current path is `TakiPortal / Angular conf / Advocacy`.

The main content area is titled "Management de mes étudiants". It displays a list of students:

- Alain Térieur, alainterieur@gmail.com, with a "Détails" button.
- Alex Térieur, alexterieur@gmail.com, with a "Détails" button and a small crown icon next to the name.
- Cara Melmou, caramelmou@gmail.com, with a "Détails" button.
- Jean Peupu, jeanpeupu@gmail.com, with a "Détails" button.
- Lara Tatouille, laratatouille@gmail.com, with a "Détails" button.

A modal dialog is open, titled "Sélectionner l'étudiant du mois :". It contains a list of students with radio buttons:

- Alain
- Alex
- Cara
- Jean
- Lara

The browser's developer tools are open on the right side, showing the DOM structure and styles applied to the page. The "Elements" tab is active, displaying the HTML code and CSS rules. The "Console" tab is also visible.

At the bottom of the screen, there is a video player interface showing a play button, volume control, and a progress bar indicating the video is at 1:09 / 1:16.

TrackBy

Take Away

Définition



fonction facultative de *ngFor



suivre la modification des éléments dans un itérable

TrackBy

Take Away

Définition



fonction facultative de *ngFor



suivre la modification des éléments dans un itérable

Avantage



performance sur des listes longues

TrackBy

Take Away

Définition



fonction facultative de *ngFor



suivre la modification des éléments dans un itérable

Avantage



performance sur des listes longues





unsubscribe

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```

Unsubscribe

user.component.ts

```
@Component({
  selector: 'user',
  templateUrl: './user.component.html',
})
export class UserComponent implements OnDestroy {
  userLanguage: string;
  subscription: Subscription;

  constructor(languageService: LanguageService) {
    this.subscription = languageService
      .observeLanguageChange()
      .subscribe((lang) => (this.userLanguage = lang));
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}
```

user.component.html

```
<div>{ { userLanguage } }</div>
```



Observable = fuite mémoire



Observable = fuite mémoire



les souscriptions subsistent



Observable = fuite mémoire



les souscriptions subsistent



penser à unsubscribe()

Unsubscribe



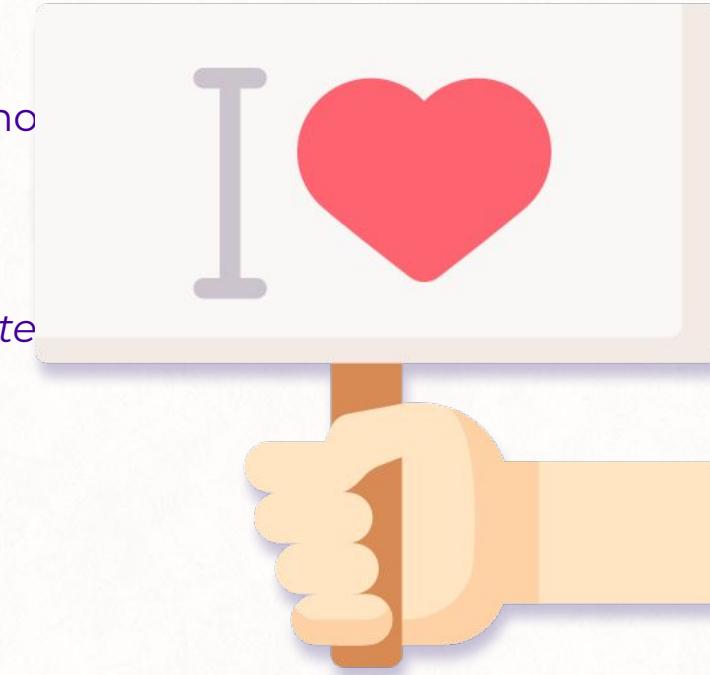
Observable = fuite mémo



les souscriptions subsiste

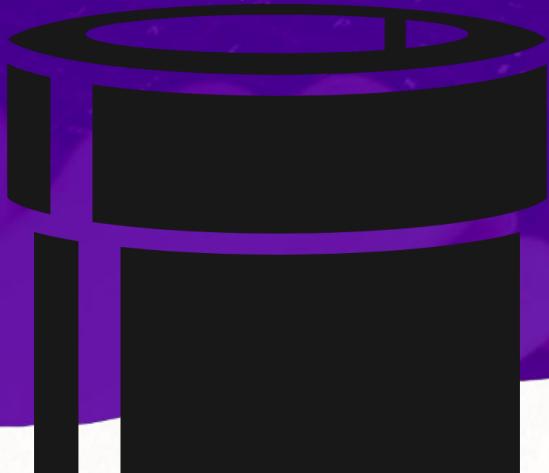


penser à unsubscribe





Le pipe



Le pipe

Qu'est ce que c'est ?

Définition

- fonction exclusivement côté HTML
- une entrée
- une sortie transformée

```
<div>{{ user.name | uppercase }} </div>
```

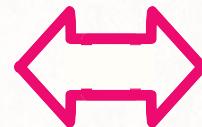
Le pipe

Qu'est ce que c'est ?

Définition

- fonction exclusivement côté HTML
- une entrée
- une sortie transformée

```
<div>{{ user.name | uppercase }} </div>
```



```
function uppercase(username: string): string {  
  return username.toUpperCase()  
}
```

Le pipe

Qu'est ce que c'est ?

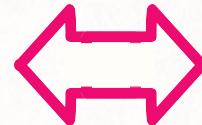
Définition

fonction exclusivement côté HTML

une entrée

une sortie transformée

```
<div>{{ user.name | uppercase }} </div>
```



```
function uppercase(username: string): string {  
  return username.toUpperCase()  
}
```



dans *CommonModule* : uppercase mais pas que...

Aa

lowercase

Aa

lowercase



date



Aa

lowercase



date



currency



Aa

lowercase



date



decimal



currency



Aa

lowercase



date



async



decimal



currency

Les pipes | **async**



utilisable sur des observables



subscribe implicitement

user.component.ts

```
user$: Observable<User>;
```

user.component.html

```
<div>{{ user$ | async }}</div>
```



pas besoin de *unsubscribe*

Les pipes

Avant | async

user.component.ts

```
export class UserComponent implements OnDestroy {  
  userLanguage: string;  
  subscription: Subscription;  
  
  constructor(languageService: LanguageService) {  
    this.subscription = languageService  
      .observeLanguageChange()  
      .subscribe((lang) => (this.userLanguage = lang));  
  }  
  
  ngOnDestroy(): void {  
    this.subscription.unsubscribe();  
  }  
}
```

user.component.html

```
<div>{{ userLanguage }}</div>
```

Les pipes

Avec | async

user.component.ts

```
export class UserComponent implements OnDestroy {  
  userLanguage$ = this.languageService  
    .observeLanguageChange()  
  
  constructor(private languageService: LanguageService) {}  
}
```

user.component.html

```
<div>{{ userLanguage$ | async }}</div>
```

Les pipes

Avec | async

user.component.ts

```
export class UserComponent implements OnDestroy {  
  userLanguage$ = this.languageService  
    .observeLanguageChange()  
  
  constructor(private languageService: LanguageService) {}  
}
```

user.component.html

```
<div>{{ userLanguage$ | async }}</div>
```

Les pipes

Avec | async

user.component.ts

```
export class UserComponent implements OnDestroy {  
  userLanguage$ = this.languageService  
    .observeLanguageChange()  
  
  constructor(private languageService: LanguageService) {}  
}
```

user.component.html

```
<div>{ { userLanguage$ | async } }</div>
```

Les pipes

Avec | async

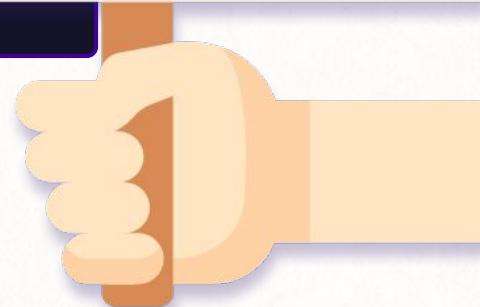
user.component.ts

```
export class UserComponent implements OnInit {
  userLanguage$ = this.languageService
    .observeLanguageChange()

  constructor(private languageService: LanguageService) {}
}
```

user.component.html

```
<div>{{ userLanguage$ | async } }</div>
```



Les pipes

Utiliser le | async

user.component.ts

```
user$: Observable<User>;
```

user.component.html

```
<span>(user$ | async)?.firstname</span>
<span>(user$ | async)?.lastname</span>
```

Les pipes

Utiliser le | async

user.component.ts

```
user$: Observable<User>;
```

user.component.html

```
<span>(user$ | async)?.firstname</span>
<span>(user$ | async)?.lastname</span>
```



2 souscriptions créées

Les pipes

Utiliser le | async avec *ngIf as

user.component.ts

```
user$: Observable<User>;
```

user.component.html

```
<ng-container *ngIf="user$ | async as user">
  <span>{{ user.firstname }}</span>
  <span>{{ user.lastname }}</span>
</ng-container>
```

Les pipes

Utiliser le | async avec *ngIf as

user.component.ts

```
user$: Observable<User>;
```

user.component.html

```
<ng-container *ngIf="user$ | async as user">
  <span>{{ user.firstname }}</span>
  <span>{{ user.lastname }}</span>
</ng-container>
```



souscription unique

Les pipes

Utiliser le | async avec *ngIf as

user.component.ts

```
user$: Observable<User>;
```



souscription unique

```
<ng-container *ngIf="user$ | async as user">
  <span>{{ user.firstName }}</span>
  <span>{{ user.lastName }}</span>
</ng-container>
```

use





Les Resolvers

Resolver

Définition



type de service



résoudre des données

Resolver

Définition



type de service



résoudre des données



Resolver

Démo en slow 3G

The screenshot shows a web browser window with a pink header bar containing the URL `localhost:4200`. Below the header, there are five student profiles listed in two rows:

Student Name	Email	Action
Alain Térieur	alainterieur@gmail.com	
Alex Térieur	alexterior@gmail.com	Détails
Cara Melmou	caramelmou@gmail.com	Détails
Jean Peupu	jeanpeupu@gmail.com	Détails
Lara Tatouille	laratatouille@gmail.com	Détails

Below the profiles, a dropdown menu is open with the following options:

- Disabled
- No throttling
- Slow 3G** (highlighted)
- Fast 3G
- Offline
- Custom
- Add...

The Network tab in the developer tools shows a list of requests and a waterfall chart. The requests include local files like `local.js`, CSS files like `css2?f...`, and various JavaScript files like `main.js`. The waterfall chart shows the duration of each request, with many requests taking between 2 and 4 seconds.

At the bottom of the browser window, there is a video player interface showing a progress bar from 0:10 to 3:14, a play button, and volume controls.

Resolver

Démo sans resolver

The screenshot shows a browser window with the URL `localhost:4200/student/4611b784-d091-4b37-a05f-f8bdda508597`. The title bar indicates the page is titled "Management de mes étudiants". The main content area shows a large pink circular loading icon and the text "Chargement...". The browser's address bar also lists "TakiPortal", "Angular conf", and "Advocacy". To the right of the browser window is the "Network" tab of the developer tools. This tab displays a timeline of network requests. The requests listed include:

Name	S.	T.	Init..	S.	T.	Waterfall
local...	3...	d...	Ot...	2...	2...	█
css2if...	2...	s...	(in...	1...	0...	█
icon?f...	2...	s...	(in...	1...	0...	█
KFOm...	2...	f...	cs...	1...	0...	█
styles....	3...	s...	(in...	2...	2...	█
runtime...	3...	s...	(in...	2...	2...	█
polyfill...	3...	s...	(in...	2...	2...	█
vendor.js	3...	s...	(in...	2...	2...	█
main.js	3...	s...	(in...	2...	2...	█
styles.js	3...	s...	(in...	2...	2...	█
students	2...	x...	zo...	1...	1...	█
ws	1...	w...	We...	0...	P...	█
crown...	3...	p...	pla...	2...	3...	█
4611b...	(...	x...	zo...	0...	P...	█
details...	(...	x...	zo...	0...	P...	█
back...	(...	p...	pla...	0...	P...	█

The timeline shows several requests starting around 10,000ms and continuing up to 40,000ms. The developer tools interface includes various filters and settings at the top, and a footer with performance metrics at the bottom.

Resolver

Démo sans resolver

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:4200/student/4611b784-d091-4b37-a05f-f8bdda508597
- Page Title:** Management de mes étudiants
- Content Area:** Displays a student profile for "Jean".
 - Image: A pink cartoon head icon.
 - Details:
 - Prénom : Jean
 - Nom : Peupu
 - Age : 30 ans
 - Mail: jeanpeupu@gmail.com
 - Message: "Bonjour Camille, j'ai fini la formation Java !"
 - Text input field: "Ecrire un message..."
 - Send button: "Envoyer"
- Network Tab:** Shows network requests for the page.

Name	S.	T.	Init.	S.	T.	Waterfall
localhost...	3...	d...	Ot...	2...	2...	
css2?f...	2...	s...	(in...	(...	0...	
icon?f...	2...	s...	(in...	(...	0...	
KFOm...	2...	f...	cs...	(...	0...	
styles...	3...	s...	(in...	2...	2...	
runtime...	3...	s...	(in...	2...	2...	
polyfill...	3...	s...	(in...	2...	2...	
vendor.js	3...	s...	(in...	2...	2...	
main.js	3...	s...	(in...	2...	2...	
styles.js	3...	s...	(in...	2...	2...	
students	2...	x...	zo...	1...	1...	
ws	1...	w...	We...	0...	P...	
crown...	3...	p...	pla...	2...	3...	
4611b...	2...	x...	zo...	4...	2...	
details...	2...	x...	zo...	4...	2...	
back-...	3...	p...	pla...	2...	2...	
boy.png	3...	p...	pla...	2...	2...	

At the bottom of the browser window, there is a video player interface showing a progress bar at 0:28, a volume icon, and a timestamp of 0:27 / 3:14.

Resolver

Démo ajout d'un resolver

The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER** view: Shows the project structure with files like `details.component.ts`, `student.resolver.ts`, and `app-routing.module.ts`.
- CODE EDITOR**: The `student.resolver.ts` file is open, containing the following code:

```
1 import { Injectable } from '@angular/core';
2 import {
3   Router,
4   Resolve,
5   ActivatedRouteSnapshot
6 } from '@angular/router';
7 import { Student } from '@core/model/student';
8 import { StudentService } from '@core/service/student.service';
9 import { Observable, of } from 'rxjs';
10
11 @Injectable({
12   providedIn: 'root'
13 })
14 export class StudentResolver implements Resolve<Student> {
15   constructor(private studentService: StudentService){}
16   resolve(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): Observable<Student> {
17     const id = route.params['id'];
18     return this.studentService.getStudent(id);
19   }
20 }
```

- TERMINAL**: The terminal shows the command being run: `ng generate resolver student`. A red box highlights the output of this command.
- STATUS BAR**: Shows the current file is `master*`, the line is `Ln 17, Col 32`, and the character is `LF`.

A yellow brushstroke is visible on the right side of the screen, and the number **67** is in the bottom right corner.

Resolver

Démo ajout resolver dans le module

The screenshot shows the VS Code interface with the Explorer, Editor, Problems, Output, Debug Console, and Terminal tabs. The Explorer sidebar shows the project structure with files like details.component.ts, student.resolver.ts, and app-routing.module.ts. The Editor tab displays the code for app-routing.module.ts, specifically the routes configuration. A red box highlights the 'resolve' section of a route definition:

```
    const routes: Routes = [
      {
        path: '',
        pathMatch: 'full',
        component: HomeComponent,
      },
      {
        path: 'student/:id',
        component: DetailsComponent,
        resolve: [
          student: StudentResolver,
        ],
      },
    ];
  
```

The Problems, Output, and Terminal tabs show the command-line history, including the generation of the resolver file:

```
MacBook-Air-de-Raphael:resolver cpillot$ MacBook-Air-de-Raphael:resolver cpillot$ MacBook-Air-de-Raphael:resolver cpillot$ MacBook-Air-de-Raphael:resolver cpillot$ ng generate resolver student
CREATE src/core/resolver/student.resolver.spec.ts (370 bytes)
CREATE src/core/resolver/student.resolver.ts (402 bytes)
MacBook-Air-de-Raphael:resolver cpillot$
```

The bottom status bar indicates the file is a TypeScript file (TypeScript) and has been prettified (Prettier).

Resolver

Démo avant utilisation resolver dans le composant

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists project files and folders. In the center is the code editor area displaying `details.component.ts`. The code defines a `DetailsComponent` that uses Observables for `student$` and `details$`. It also includes logic to fetch student and details data from services based on the `studentId` parameter. At the bottom, the terminal window shows the command `ng generate resolver student` being run, followed by the creation of two files: `student.resolver.spec.ts` and `student.resolver.ts`.

```
src > app > details > details.component.ts > DetailsComponent > ngOnInit > studentId
10 selector: 'app-details',
11 templateUrl: './details.component.html',
12 styleUrls: ['./details.component.scss'],
13 })
14 export class DetailsComponent implements OnInit {
15   student$: Observable<Student>;
16   details$: Observable<Details>;
17
18   constructor(
19     private route: ActivatedRoute,
20     private studentService: StudentService,
21     private detailsService: DetailsService,
22   ) {}
23
24   ngOnInit(): void {
25     const studentId = this.route.snapshot.params['id'];
26     this: this
27     this.student$ = this.studentService.getStudent(studentId);
28     this.details$ = this.detailsService.getDetailsFromStudent(studentId);
29   }
30 }
31
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
MacBook-Air-de-Raphael:resolver cpillot$ ng generate resolver student
CREATE src/core/resolver/student.resolver.spec.ts (370 bytes)
CREATE src/core/resolver/student.resolver.ts (402 bytes)
MacBook-Air-de-Raphael:resolver cpillot$
```

node
bash resolv...

69

Resolver

Démo après utilisation resolver dans le composant

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `details.component.ts`, `student.resolver.ts`, and `app-routing.module.ts`.
- Code Editor:** Displays the `details.component.ts` file, which includes code for a `DetailsComponent` that uses a `StudentService` and a `DetailsService` via a resolver.
- Terminal:** Shows the command line output of running `ng generate resolver student`, creating two new files: `student.resolver.spec.ts` and `student.resolver.ts`.
- Status Bar:** Shows the current branch is `master*`, the file is `details.component.ts`, and the status bar indicates the code was Blame Camille Pillot (23 days ago) at Line 27, Column 74, with 2 spaces, using UTF-8 encoding.

Resolver

Démo avec resolver données directement affichées

The screenshot shows a browser window with the URL `localhost:4200/student/4611b784-d091-4b37-a05f-f8bdda508597`. The page title is "Management de mes étudiants". On the left, there is a sidebar with the following information:

- Retour
- Prénom : Jean
- Nom : Peupu
- Age : 30 ans
- Mail: jeanpeupu@gmail.com

In the center, there is a large red circular loading icon with the text "Chargement..." below it.

The right side of the browser window shows the developer tools Network tab. The timeline at the top indicates a slow 3G connection. The list of requests includes:

Name	S.	T.	Init.	S.	T.	Waterfall
localhost...	3...	d...	2...	2...	...	green bar
css2?f...	2...	s...	(in...	(in...	3...	blue bar
icon?f...	2...	s...	(in...	(in...	3...	blue bar
styles....	3...	s...	(in...	(in...	2...	green bar
runtime...	2...	s...	(in...	(in...	7...	green bar
polyfill...	3...	s...	(in...	(in...	2...	green bar
vendor.js	3...	s...	(in...	(in...	2...	green bar
main.js	2...	s...	(in...	(in...	5...	green bar
styles.js	3...	s...	(in...	(in...	2...	green bar
KFOM...	2...	f...	cs...	(in...	0...	blue bar
ws	1...	w...	We...	0...	2...	grey bar
students	2...	x...	zo...	1...	2...	grey bar
favicon...	3...	v...	Ot...	2...	2...	grey bar
crown...	3...	p...	pla...	2...	2...	grey bar
ws	1...	w...	We...	0...	2...	grey bar
4611b...	2...	x...	zo...	4...	2...	grey bar
details...	(...)	x...	zo...	0...	P...	grey bar
back-...	(...)	p...	pla...	0...	P...	grey bar
boy.png	(...)	p...	pla...	0...	P...	grey bar

At the bottom of the browser window, there are video controls: play, volume, and timestamp (3:07 / 3:14). The status bar at the bottom right shows: 20 requests, 61.9 kB transferred, 4.2 MB resources, and a finish time of 15:10:5.

Resolver

Désavantage



Le *resolver* empêche l'affichage
TOTAL de la page

Resolver

Exemple OVH mauvais UX

The screenshot shows the OVHcloud Public Cloud dashboard. At the top, there are navigation links: Tableau de bord, Bare Metal Cloud, Hosted Private Cloud, Public Cloud (selected), Web Cloud, Télécom, Sunrise, and Marketplace. A language switch to Français and a user icon are also at the top right.

The main content area is divided into several sections:

- Compute:** Includes Compute Instances, Storage Block Storage, Object Storage, Cloud Archive, Cold Archive, Databases, Volume Snapshot, and Instance Backup. Some items have "Beta" or "New" status indicators.
- Network:** Includes Private Network, Public IPs, and Gateway.
- Containers & Orchestration:** Includes Load Balancer, Managed Kubernetes Service, and Managed Private Registry.
- Facturation:** Shows a promotional code, Credit provisioning, and a balance of **0.00 €**. It also indicates the offer is valid until February 1, 2023, at 01:00.
- Documentation:** Links to Getting Started, L'essentiel pour commencer, and Public Cloud Se familiariser.
- Instances:** Links to Create an instance, Create a cluster Kubernetes, Create an Object Storage container, Create a Block Storage volume, Create a database, Create an AI Notebook, and Manage vRack private networks.
- Communauté:** Links to the Roadmap, Découvrez et participez, and Communauté Discutez sur Discord.



Chargement en cours...

Resolver

Exemple Twitter bon UX

The image shows a screenshot of the Twitter mobile application interface. At the top left is the Twitter logo. The main content area displays a user profile for 'takima' (@takimafr). The profile picture is a circular image of a group of people. The bio reads: 'Takima is a team of experts developing amazing #tech solutions with ❤️ in Paris. #StartupFactory : @jawglo @GatlingTool'. Below the bio, it says 'A rejoint Twitter en novembre 2013' and shows '236 abonnements' and '367 abonnés'. A blue 'Tweeter' button is at the bottom of the profile section. To the right of the profile are tabs for 'Tweets', 'Réponses', 'Médias', and 'J'aime'. Below these tabs is a 'Messages' button. The main feed area shows a grid of thumbnail images from other users, followed by sections for 'Vous pourriez aimer' (with profiles for Olivier Huber and Camille Pillot) and 'Tendances : France'.

Vidéo OVH chargement très
long au démarrage

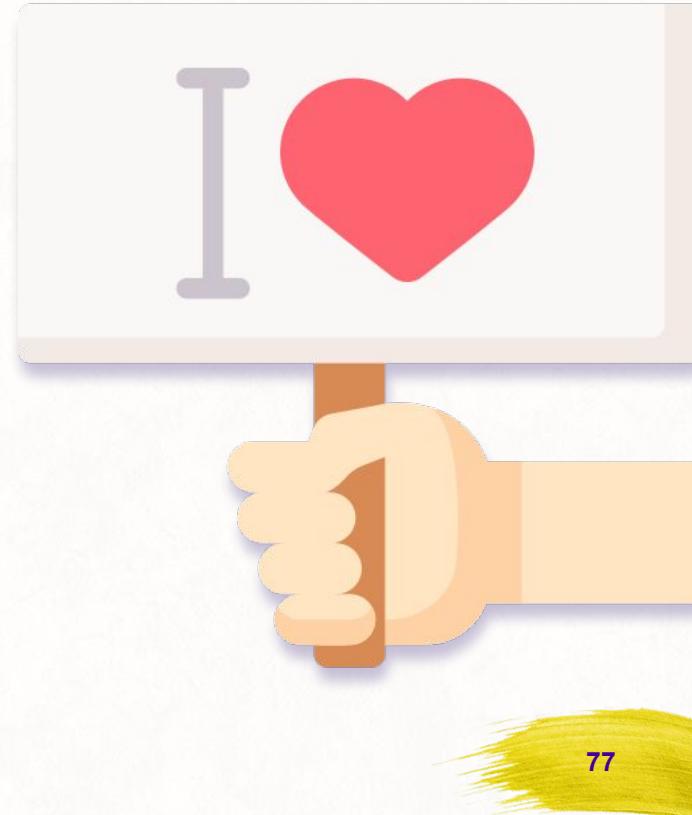
Vidéo twitter chargement
rapide au démarrage car
resolver + loader

Resolver

Expérience utilisateur



Resolver + loader





Pour résumer...

Résumé

Les alias

```
import { Banner } from  
'../../../../core/models/model';
```



```
import { Banner } from '@core/models/model';
```

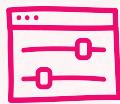
Résumé

trackBy

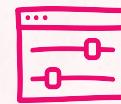


```
<div *ngFor="user of users; trackBy: trackByFn">  
    <span>{ { user.firstname } }</span>  
</div>
```

Outils navigateur



paint flashing



limitation de bande
passante

Résumé

unsubscribe



.subscribe()



.unsubscribe()



Résumé

| async

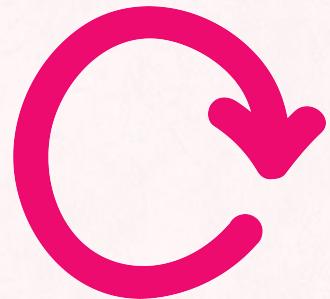
| async



*ngIf as

Résumé

Resolvers



précharge les données



à utiliser selon le cas d'usage

Merci !



Camille Pillot



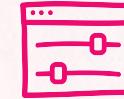
@CamiPillotDev



alias



trackBy



outils navigateur



unsubscribe



| async



resolvers