Migrer ses applications AngularJs vers Angular en dou(c|l)eur avec l'approche Hybride

Version de travail

THIBAUD DESTOUCHES
JULES DELARUE

The world is how we shape it*

* Le monde est tel que nous le façonnons.

sopra steria

Qui sommes –nous ?

Sopra Steria, un des leaders européens de la Tech

Un Groupe européen de conseil, de services numériques & d'édition de solutions

47 000
collaborateurs

4,7 Md€
CA 2021

Top 5
des acteurs européens

~30
pays

Un modèle indépendant
Reposant sur une vision long terme et la performance économique

Une culture d'entrepreneur
Agilité, autonomie de décision des managers, sens du collectif et respect de l'autre

Un groupe centré sur l'humain
Politique RH basée sur la responsabilité, la confiance et le développement des compétences

Le partenaire de confiance
Posture d'écoute, de convictions et de proximité avec nos clients et partenaires

Le centre de services APS

Des communautés d'experts au service des projets

SQUAD Architecture & Expertise (ETAPS)

36 Nombre de personnes (20 Architectes et 16 experts techniques) **dédiées** au monde de **l'Assurance et de la Protection Sociale**

La participation à des **conférences réputées** et la restitution des informations les plus marquantes / utiles auprès des équipes

Une **animation trimestrielle** (plénière) de présentation de nouvelles solutions, de partage de bonnes pratiques, de retour d'expérience.

Communauté Sécurité

Mise en place de différents systèmes dont des modules e-learning pour faire partager les bonnes pratiques en terme de **sécurité et d'anti- piratage** et l'obtention d'une certification **ISO-27001**

Communauté Industrialisation

Cellule autour de **l'industrialisation** de l'ensemble des activités, notamment au travers de la DEP (Digitale Enable Platform), basée sur la solution Cloud Microsoft AZURE

Communauté BA / Testing

21 Nombre de personnes au sein de la Communauté des Business Analyst **dédiées** au monde de **l'Assurance et de la Protection Sociale**

Formations fondamentales métiers, **Spécification** (capture et formalisation des exigences) et **Acceptance** (élaboration, déroulement et analyse de la stratégie de Test)

De quoi qu'on parle ?

01 **Un zeste de contexte, avec un soupçon d'enjeu**

02 **Démontrer la faisabilité – les prérequis**

03 ***Live coding partie 1* - hybridation de l'application PhoneCat (cohabitation simple)**

04 **Démontrer la faisabilité – en vrai (on ne part pas d'une appli clean)**

05 **Etat du projet à ce jour**

06 **Les next steps (avec encore du livecoding ?!)**

01

Contexte

Contexte et Enjeux

- **Google a abandonné le support d'AngularJS (version 1.8.x) depuis le 31/12/2021. Ce qui implique qu'il n'y aura plus de correctifs ou d'évolutions pour le framework même en cas de vulnérabilités critiques de sécurité. Pour ces raisons, il est recommandé de migrer vers le framework Angular (version 2+).**

- **A noter qu'en complément, le framework Angular permet :**
  - D'accéder à de nouvelles fonctionnalités/modules qui permettront aux applications de continuer à être alignée avec le marché.
  - De réaliser les prochaines montée de version de façon simple car Google a mis en place dans les version 2+ un outil de migration « simplifiée » indiquant les principales évolutions (https://update.angular.io/)

- **Une montée de version vers Angular est donc préconisée pour des raisons de :**
  - Sécurité
  - Performance
  - Compatibilité

Genèse

Pourquoi un framework hybride ?

- **Sopra Steria développe pour 2015 le système de gestion d'un client du monde de la protection sociale en AngularJS**
  - 350+ écrans développés
- **Impossibilité de faire une migration en « one shot » d'un périmètre aussi grand**
- **Choix de procéder à une migration de l'exhaustivité du périmètre de manière itérative sur 18 mois et 13 lots**
  - Pas de rupture de service
  - Pas de gel fonctionnel de l'ensemble du périmètre
  - Des mises en production successives à chaque fin de lot

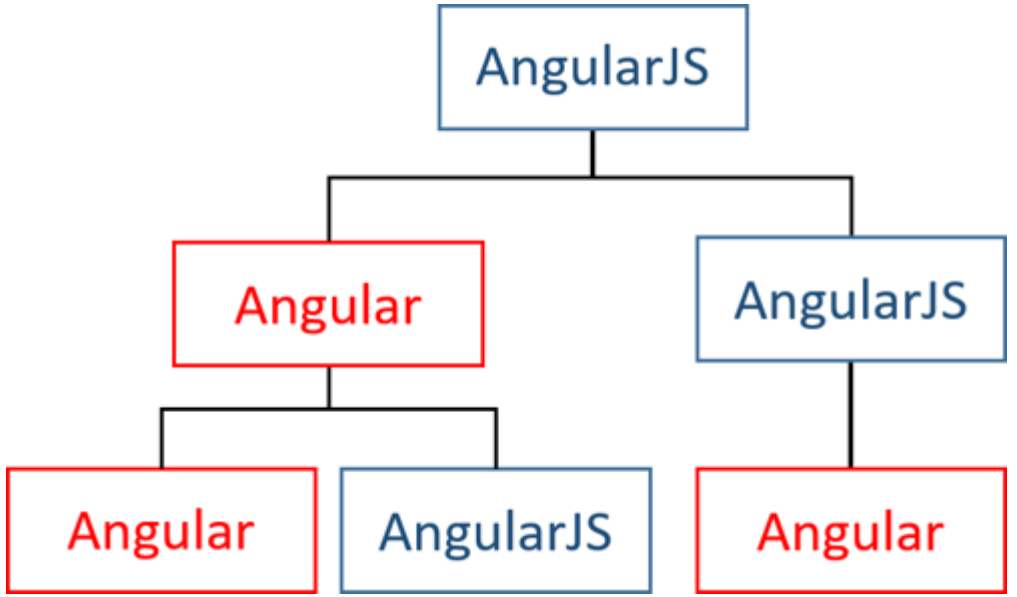
02

Faisabilité

LES PRÉREQUIS

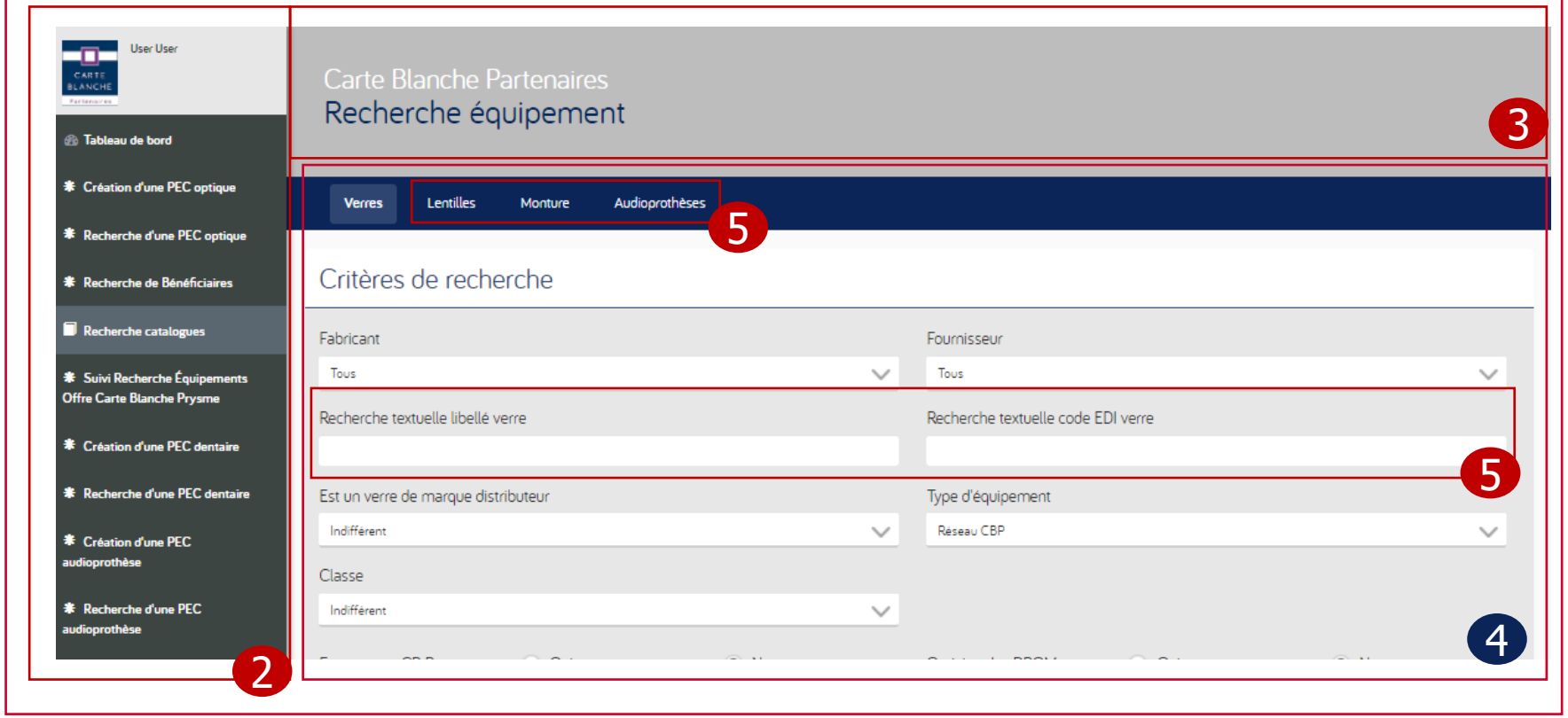
Principe de base

- **On souhaite pouvoir faire cohabiter les écrans AngularJS et Angular**
  - Transparent pour l'utilisateur
  - Sans produire une usine à gaz
  - Permettant de migrer les écrans de manière itérative
- **On souhaite se placer dans l'état de l'art d'Angular sans casser le legacy**
  - Du code Angular propre et maintenable à la fin de la migration
  - En utilisant les bonnes pratiques
- **On éprouve nos hypothèse avec un PoC**



Proof of Concept

1



- La nouvelle version d'Angular se charge d'assurer la rétrocompatibilité 1
- Le menu et l'entête sont toujours sur la version historique 2 3
- Le contenu de la page est sur la nouvelle version 4 mais ça ne l'empêche pas de réutiliser des éléments de la version actuelle 5

— Nouveau composant (Angular)

— Composant de la version actuelle (AngularJS)

03

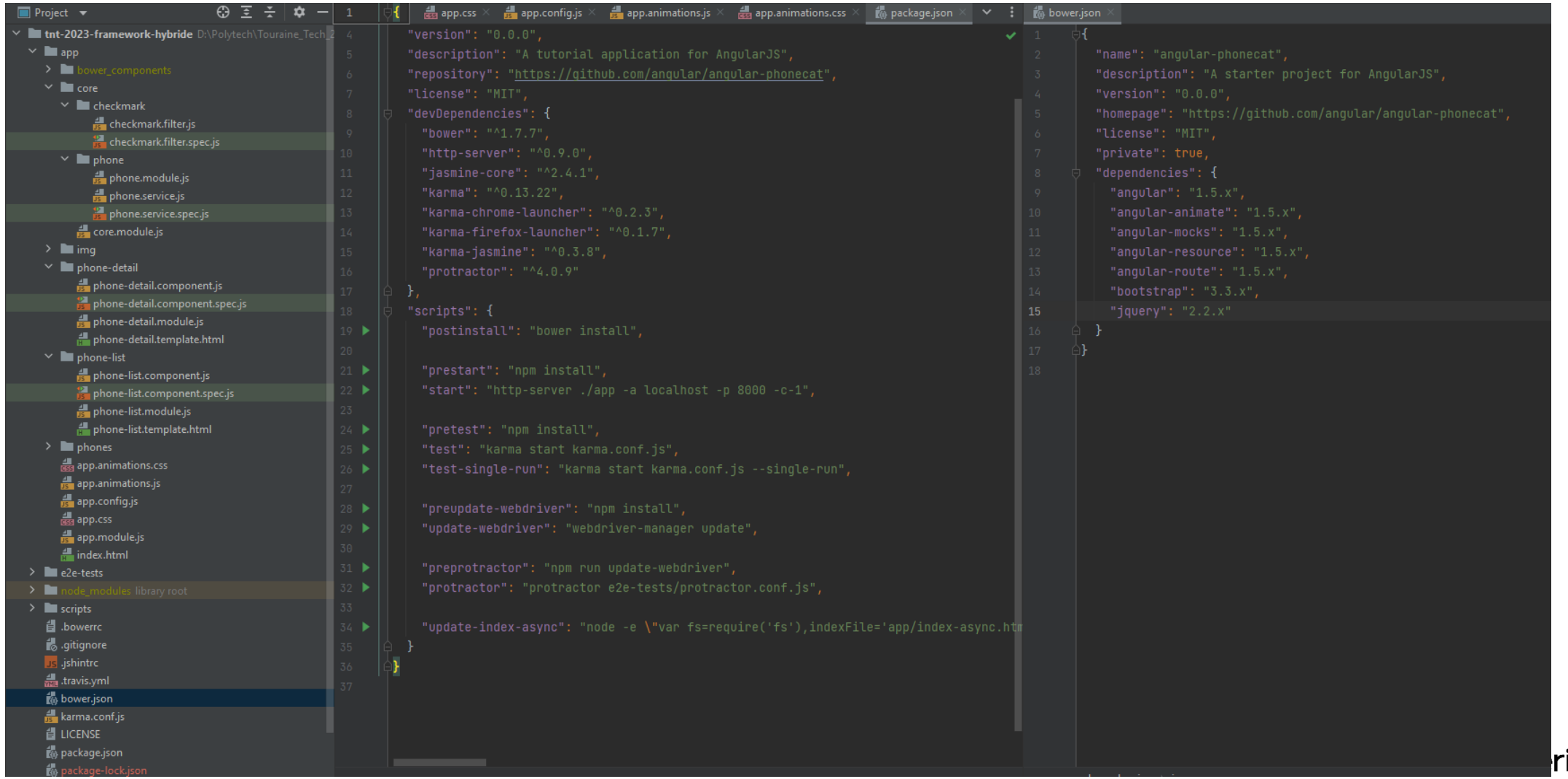
Livecoding

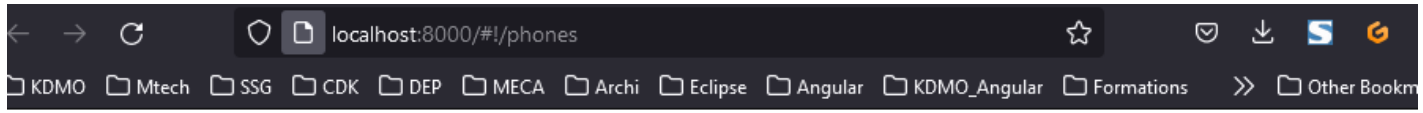
L’application à migrer

https://github.com/angular/angular-phonecat/commits/1.5-snapshot

- **Tour de l’appli**
- **Init nouveau projet angular 13**
- **Recopier les sources AngularJs**
- **Bootstrap IT**
- **Voilà c’est hybride**

Le legacy


Project
tnt-2023-framework-hybride D:\Polytech\Touraine_Tech_2
app
bower_components
core
checkmark
checkmark.filter.js
checkmark.filter.spec.js
phone
phone.module.js
phone.service.js
phone.service.spec.js
core.module.js
img
phone-detail
phone-detail.component.js
phone-detail.component.spec.js
phone-detail.module.js
phone-detail.template.html
phone-list
phone-list.component.js
phone-list.component.spec.js
phone-list.module.js
phone-list.template.html
phones
app.animations.css
app.animations.js
app.config.js
app.css
app.module.js
index.html
e2e-tests
node_modules library root
scripts
.bowerrc
.gitignore
.jshintrc
.travis.yml
bower.json
karma.conf.js
LICENSE
package.json
package-lock.json
app.css
app.config.js
app.animations.js
app.animations.css
package.json
bower.json
"version": "0.0.0",
"description": "A tutorial application for AngularJS",
"repository": "https://github.com/angular/angular-phonecat",
"license": "MIT",
"devDependencies": {
"bower": "^1.7.7",
"http-server": "^0.9.0",
"jasmine-core": "^2.4.1",
"karma": "^0.13.22",
"karma-chrome-launcher": "^0.2.3",
"karma-firefox-launcher": "^0.1.7",
"karma-jasmine": "^0.3.8",
"protractor": "^4.0.9"
},
"scripts": {
"postinstall": "bower install",
"prestart": "npm install",
"start": "http-server ./app -a localhost -p 8000 -c-1",
"pretest": "npm install",
"test": "karma start karma.conf.js",
"test-single-run": "karma start karma.conf.js --single-run",
"preupdate-webdriver": "npm install",
"update-webdriver": "webdriver-manager update",
"preprotractor": "npm run update-webdriver",
"protractor": "protractor e2e-tests/protractor.conf.js",
"update-index-async": "node -e \"var fs=require('fs'),indexFile='app/index-async.htm
}
}
{
"name": "angular-phonecat",
"description": "A starter project for AngularJS",
"version": "0.0.0",
"homepage": "https://github.com/angular/angular-phonecat",
"license": "MIT",
"private": true,
"dependencies": {
"angular": "1.5.x",
"angular-animate": "1.5.x",
"angular-mocks": "1.5.x",
"angular-resource": "1.5.x",
"angular-route": "1.5.x",
"bootstrap": "3.3.x",
"jquery": "2.2.x"
}
}



Search:

Sort by: Newest

Motorola XOOM™ with Wi-Fi

The Next, Next Generation Experience the future with Motorola XOOM with Wi-Fi, the world's first tablet powered by Android 3.0 (Honeycomb).

MOTOROLA XOOM™

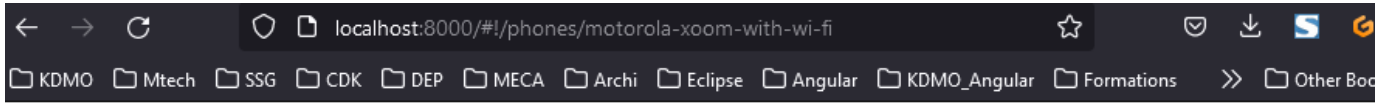
The Next, Next Generation Experience the future with MOTOROLA XOOM, the world's first tablet powered by Android 3.0 (Honeycomb).

MOTOROLA ATRIX™ 4G

MOTOROLA ATRIX 4G the world's most powerful smartphone.

Dell Streak 7

Introducing Dell™ Streak 7. Share photos, videos and movies together. It's small enough to carry around, big enough to gather around.



Motorola XOOM™ with Wi Fi

Motorola XOOM with Wi-Fi has a super-powerful dual-core processo and Android™ 3.0 (Honeycomb) — the Android platform designed specifically for tablets. With its 10.1-inch HD widescreen display, you enjoy HD video in a thin, light, powerful and upgradeable tablet.






Availability and Networks
Availability

Battery
Type
Other (mAH)
Talk Time
24 hours
Standby time (max)
336 hours

Storage and Memory
RAM
1000MB
Internal Storage
32000MB

Connectivity
Network Support
WiFi
802.11 b/g/n
Bluetooth
Bluetooth 2.1
Infrared
✗
GPS
✓

Android
OS Version
Android 3.0
UI

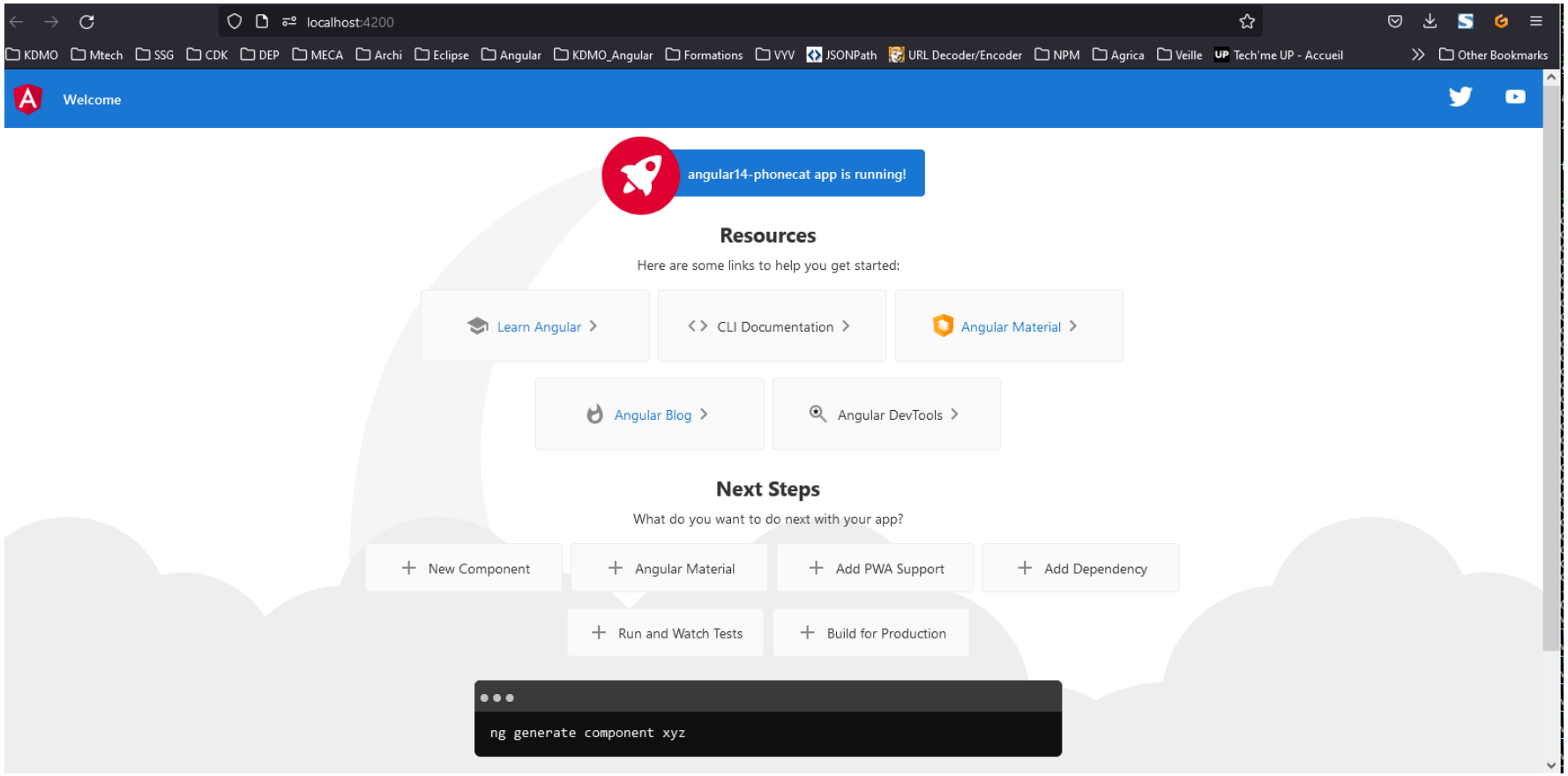
Size and Weight
Dimensions
249.1 mm (w)
167.8 mm (h)

Display
Screen size
10.1 inches
Screen resolution

Hardware
CPU
1 GHz Dual Core Tegra 2
USB

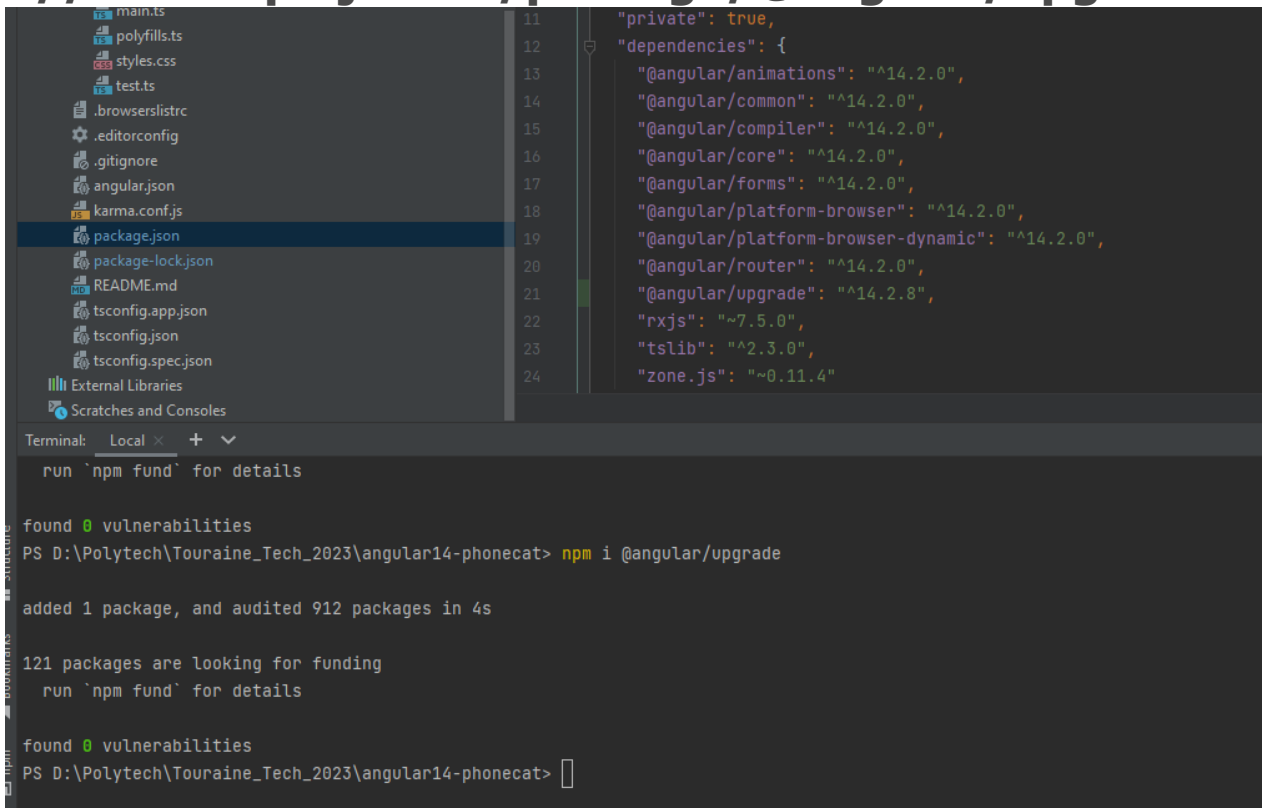
Init projet tout propre

```
jdelarue@ITEM-S95818 MINGW64 /d/Polytech/Touraine_Tech_2023
$ ng new angular14-phonecat
CREATE angular14-phonecat/angular.json (2982 bytes)
CREATE angular14-phonecat/package.json (1049 bytes)
CREATE angular14-phonecat/README.md (1071 bytes)
CREATE angular14-phonecat/tsconfig.json (863 bytes)
CREATE angular14-phonecat/.editorconfig (274 bytes)
CREATE angular14-phonecat/.gitignore (548 bytes)
CREATE angular14-phonecat/.browserslistrc (600 bytes)
CREATE angular14-phonecat/karma.conf.js (1435 bytes)
CREATE angular14-phonecat/tsconfig.app.json (287 bytes)
CREATE angular14-phonecat/tsconfig.spec.json (333 bytes)
CREATE angular14-phonecat/.vscode/extensions.json (130 bytes)
CREATE angular14-phonecat/.vscode/launch.json (474 bytes)
CREATE angular14-phonecat/.vscode/tasks.json (938 bytes)
CREATE angular14-phonecat/src/favicon.ico (948 bytes)
CREATE angular14-phonecat/src/index.html (303 bytes)
CREATE angular14-phonecat/src/main.ts (372 bytes)
CREATE angular14-phonecat/src/polyfills.ts (2338 bytes)
CREATE angular14-phonecat/src/styles.css (80 bytes)
CREATE angular14-phonecat/src/test.ts (749 bytes)
CREATE angular14-phonecat/src/assets/.gitkeep (0 bytes)
CREATE angular14-phonecat/src/environments/environment.prod.ts (51 bytes)
CREATE angular14-phonecat/src/environments/environment.ts (658 bytes)
CREATE angular14-phonecat/src/app/app.module.ts (314 bytes)
CREATE angular14-phonecat/src/app/app.component.html (23083 bytes)
CREATE angular14-phonecat/src/app/app.component.spec.ts (992 bytes)
CREATE angular14-phonecat/src/app/app.component.ts (222 bytes)
CREATE angular14-phonecat/src/app/app.component.css (0 bytes)
- Installing packages (npm)...
```


localhost:4200
KDMO
Mtech
SSG
CDK
DEP
MECA
Archi
Eclipse
Angular
KDMO_Angular
Formations
VYV
JSONPath
URL Decoder/Encoder
NPM
Agrica
Veille
Tech'me UP - Accueil
Other Bookmarks
Welcome
angular14-phonecat app is running!
Resources
Here are some links to help you get started:
Learn Angular
CLI Documentation
Angular Material
Angular Blog
Angular DevTools
Next Steps
What do you want to do next with your app?
New Component
Angular Material
Add PWA Support
Add Dependency
Run and Watch Tests
Build for Production
ng generate component xyz

Merge nouvelle app avec le legacy

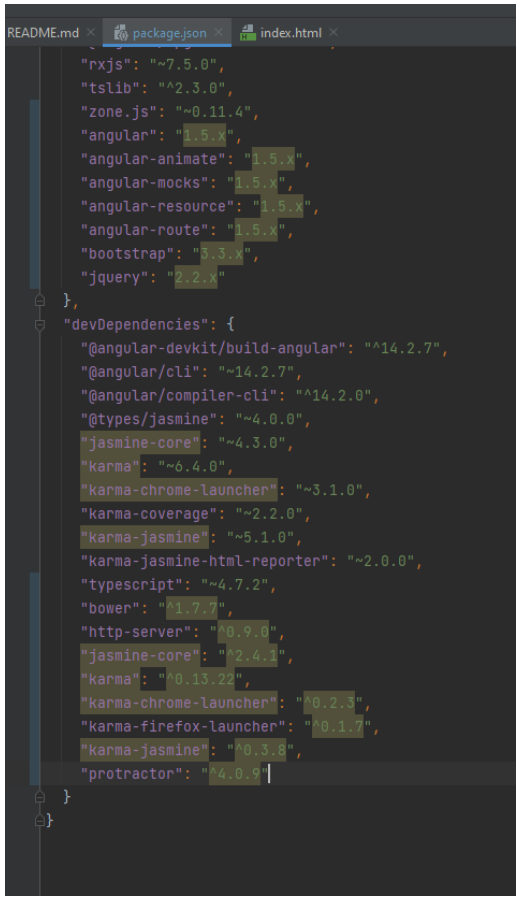
- Copier src du legacy dans un dossier legacy
- Installer https://www.npmjs.com/package/@angular/upgrade



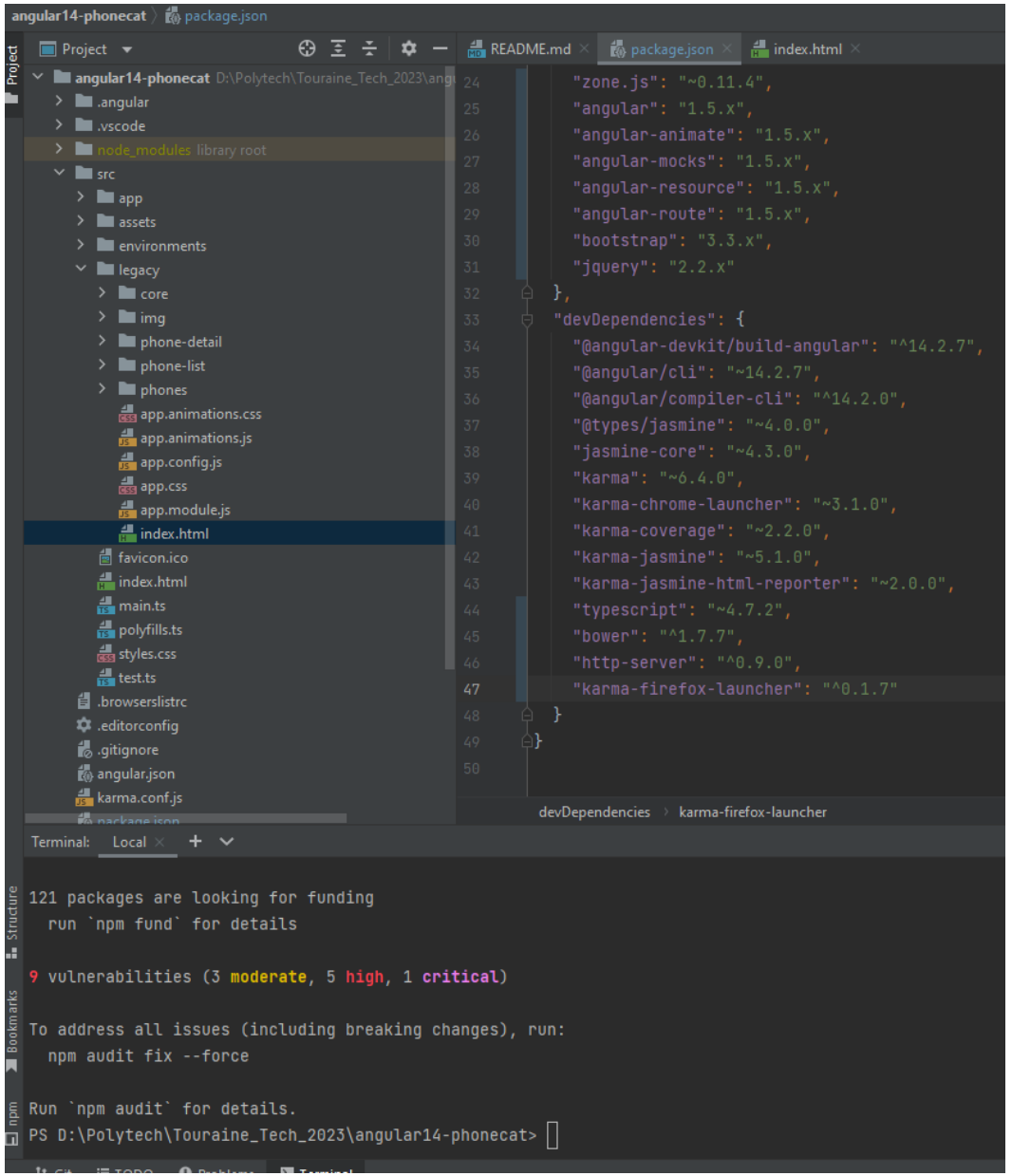
Bower et dev dependencies ?

- **Zou on déplace tout dans le package.json**
- **Que faire des doublons ?**
  - On garde la dernière version (discutable)
  - On identifie les conflits

```
ERR! ERESOLVE could not resolve
ERR!
ERR! While resolving: @angular-devkit/build-angular@14.2.7
ERR! Found: protractor@4.0.14
ERR! node_modules/protractor
ERR!   dev protractor@"^4.0.9" from the root project
ERR!
ERR! Could not resolve dependency:
ERR! peerOptional protractor@"^7.0.0" from @angular-devkit/build-angular@14.2.7
ERR! node_modules/@angular-devkit/build-angular
ERR!   dev @angular-devkit/build-angular@"^14.2.7" from the root project
ERR! to accept an incorrect (and potentially broken) dependency resolution.
ERR!
ERR! See C:\Users\jdelarue\AppData\Local\npm-cache\eresolve-report.txt for a full report.
```



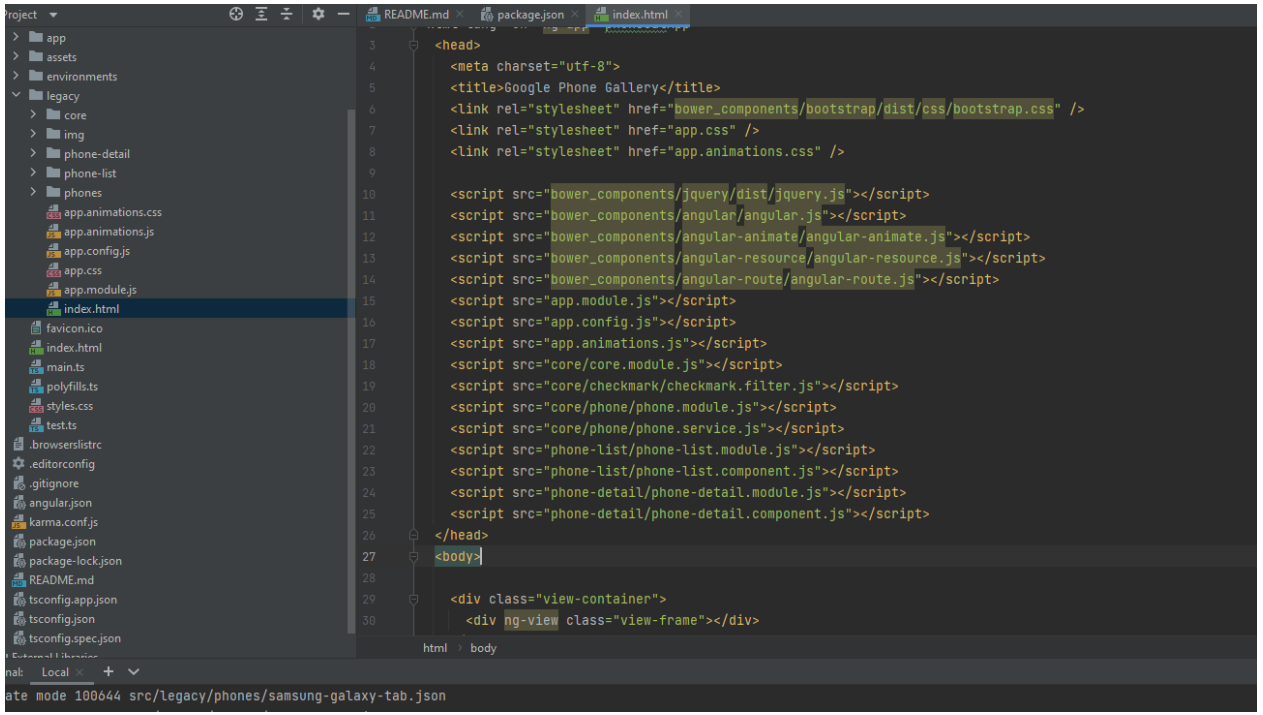
- **Est-ce vraiment utile de garder bower et http server ?**
  - Probablement pas, mais dans le doute on garde



On a tout pour faire fonctionner le legacy

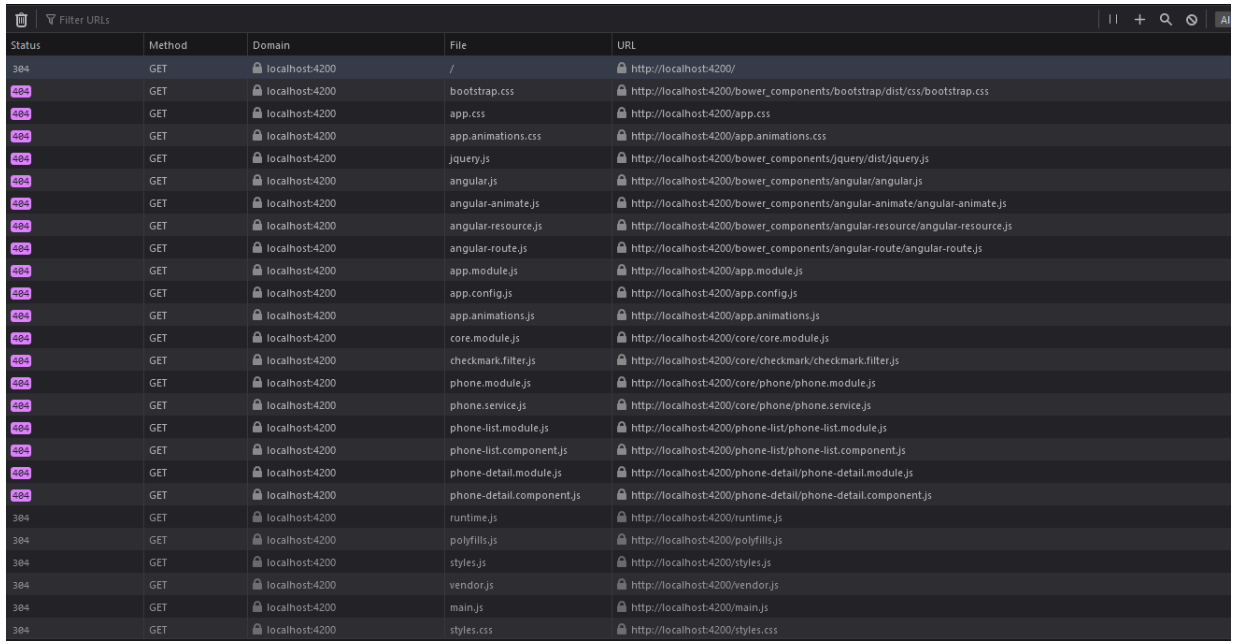
Que faire des scripts ? Et des css ?

- Soit dans le index.html angular 14 (beurk ?)
- Soit dans le angular.json (better ?)
- Soit un barrel ?



Au plus simple -> copier coller des scripts

- Que se passe il ?
- Ah c'est pas trouvé !
- Bon les scripts bower, vu que y'a plus bower c'est normal, c'est dans le node modules

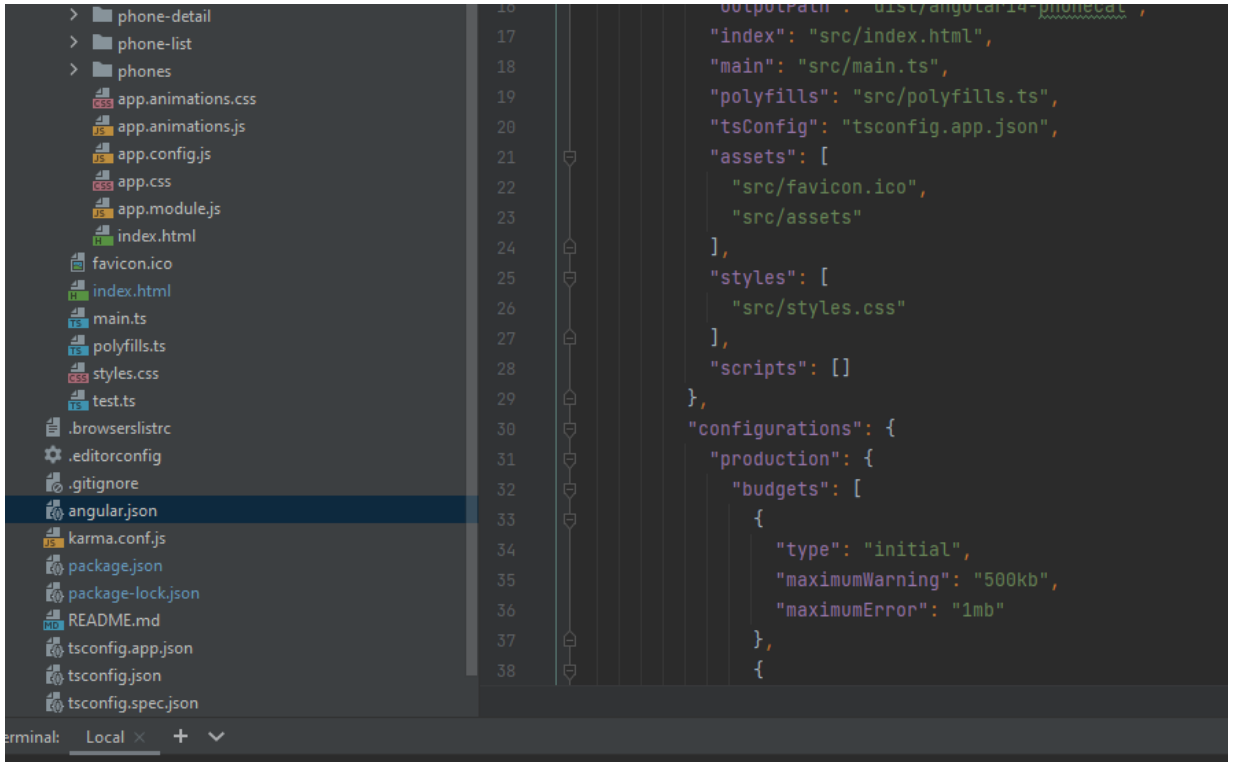


Filter URLs

Status	Method	Domain	File	URL
304	GET	localhost:4200	/	http://localhost:4200/
404	GET	localhost:4200	bootstrap.css	http://localhost:4200/bower_components/bootstrap/dist/css/bootstrap.css
404	GET	localhost:4200	app.css	http://localhost:4200/app.css
404	GET	localhost:4200	app.animations.css	http://localhost:4200/app.animations.css
404	GET	localhost:4200	jquery.js	http://localhost:4200/bower_components/jquery/dist/jquery.js
404	GET	localhost:4200	angular.js	http://localhost:4200/bower_components/angular/angular.js
404	GET	localhost:4200	angular-animate.js	http://localhost:4200/bower_components/angular-animate/angular-animate.js
404	GET	localhost:4200	angular-resource.js	http://localhost:4200/bower_components/angular-resource/angular-resource.js
404	GET	localhost:4200	angular-route.js	http://localhost:4200/bower_components/angular-route/angular-route.js
404	GET	localhost:4200	app.module.js	http://localhost:4200/app.module.js
404	GET	localhost:4200	app.config.js	http://localhost:4200/app.config.js
404	GET	localhost:4200	app.animations.js	http://localhost:4200/app.animations.js
404	GET	localhost:4200	core.module.js	http://localhost:4200/core/core.module.js
404	GET	localhost:4200	checkmark.filter.js	http://localhost:4200/core/checkmark/checkmark.filter.js
404	GET	localhost:4200	phone.module.js	http://localhost:4200/core/phone/phone.module.js
404	GET	localhost:4200	phone.service.js	http://localhost:4200/core/phone/phone.service.js
404	GET	localhost:4200	phone-list.module.js	http://localhost:4200/phone-list/phone-list.module.js
404	GET	localhost:4200	phone-list.component.js	http://localhost:4200/phone-list/phone-list.component.js
404	GET	localhost:4200	phone-detail.module.js	http://localhost:4200/phone-detail/phone-detail.module.js
404	GET	localhost:4200	phone-detail.component.js	http://localhost:4200/phone-detail/phone-detail.component.js
304	GET	localhost:4200	runtime.js	http://localhost:4200/runtime.js
304	GET	localhost:4200	polyfills.js	http://localhost:4200/polyfills.js
304	GET	localhost:4200	styles.js	http://localhost:4200/styles.js
304	GET	localhost:4200	vendor.js	http://localhost:4200/vendor.js
304	GET	localhost:4200	main.js	http://localhost:4200/main.js
304	GET	localhost:4200	styles.css	http://localhost:4200/styles.css

Oui mais

- On ne charge pas comme un sale les librairies de ton projet par du script en angular 14
- On les mets au propres dans scripts



A mi chemin…

- On a pas fait les liens bower, mais testons déjà ça
- RIP ☹

```
"styles": [
  "src/styles.css",
  "bower_components/bootstrap/dist/css/bootstrap.css",
  "src/legacy/app.css",
  "src/legacy/app.animations.css"
],
"scripts": [
  "bower_components/jquery/dist/jquery.js",
  "bower_components/angular/angular.js",
  "bower_components/angular-animate/angular-animate.js",
  "bower_components/angular-resource/angular-resource.js",
  "bower_components/angular-route/angular-route.js",
  "src/legacy/app.module.js",
  "src/legacy/app.config.js",
  "src/legacy/app.animations.js",
  "src/legacy/core/core.module.js",
  "src/legacy/core/checkmark/checkmark.filter.js",
  "src/legacy/core/phone/phone.module.js",
  "src/legacy/core/phone/phone.service.js",
  "src/legacy/phone-list/phone-list.module.js",
  "src/legacy/phone-list/phone-list.component.js",
  "src/legacy/phone-detail/phone-detail.module.js",
  "src/legacy/phone-detail/phone-detail.component.js"
]
```

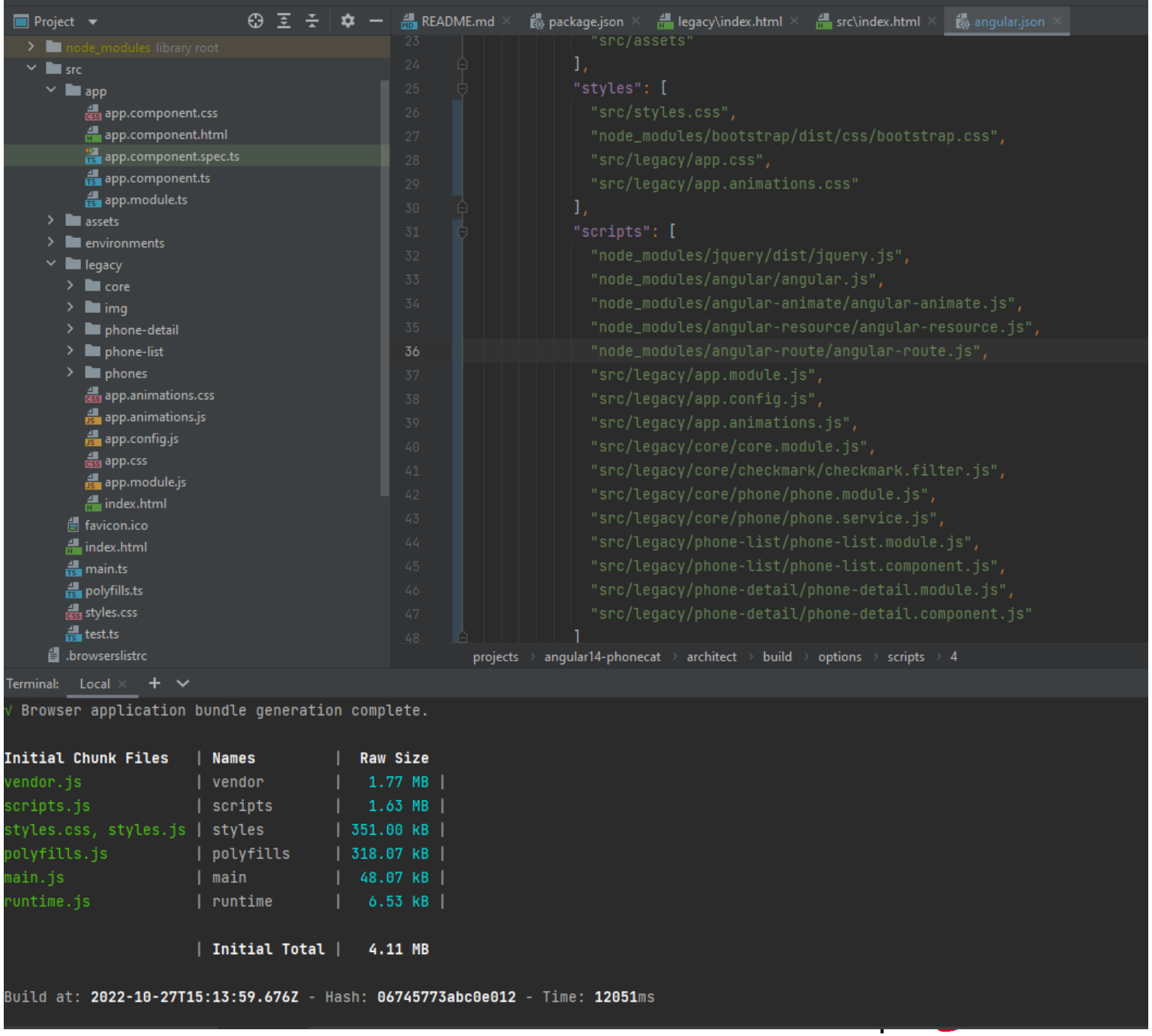
```
                    ^

Error: Script file bower_components/jquery/dist/jquery.js does not exist.
    at D:\Polytech\Touraine_Tech_2023\angular14-phonecat\node_modules\@angular-devkit\build-angular\src\webpack\utils\helpers.js:177:23
    at Array.reduce (<anonymous>)
    at globalScriptsByBundleName (D:\Polytech\Touraine_Tech_2023\angular14-phonecat\node_modules\@angular-devkit\build-angular\src\webpack\utils\helper
    at getCommonConfig (D:\Polytech\Touraine_Tech_2023\angular14-phonecat\node_modules\@angular-devkit\build-angular\src\webpack\configs\common.js:100:
PS D:\Polytech\Touraine_Tech_2023\angular14-phonecat>
```

README.md | package.json | legacy\index.html | src\index.html | angular.json

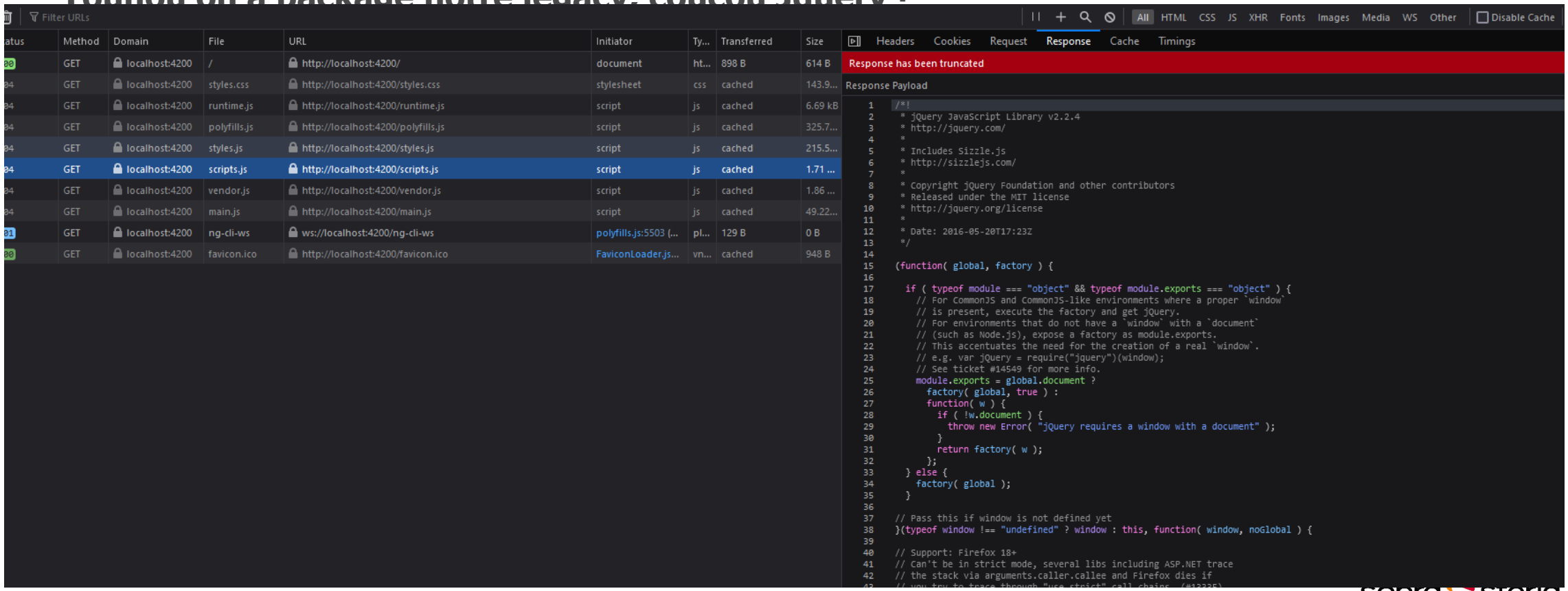
```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Angular14Phonecat</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>
<body>
  <app-root></app-root>
</body>
</html>
```

- **Avant / Après**
- **C'est dans scripts.js**

Initial Chunk Files	Names	Raw Size
vendor.js	vendor	1.77 MB
polyfills.js	polyfills	318.07 kB
styles.css, styles.js	styles	210.10 kB
main.js	main	48.07 kB
runtime.js	runtime	6.53 kB



Youhou on a packagé notre legacy, coucou Jquery !



On est bien avancé, c'est quoi la suite ?

- Là on a notre app angular 14, qui contient (au sens fichier) notre app legacy
- Là On bootstrap angular 14 (App component)

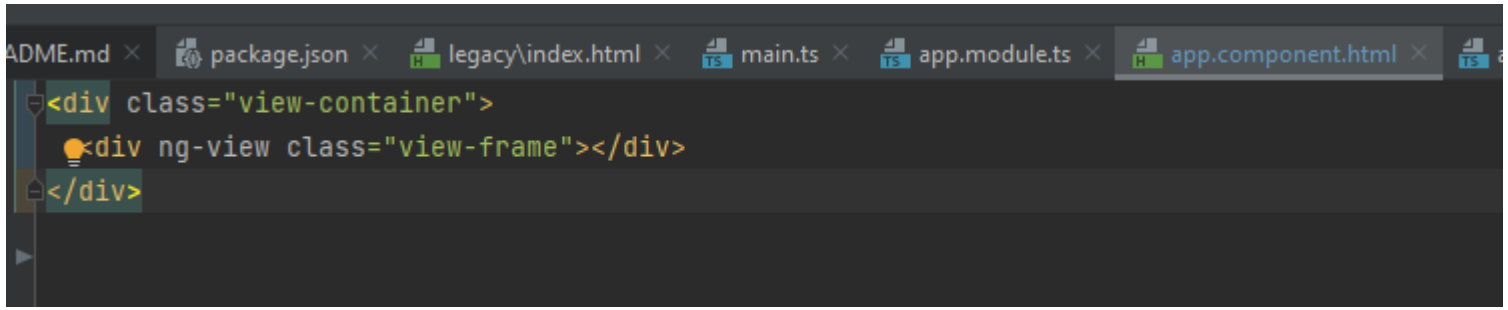
```
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

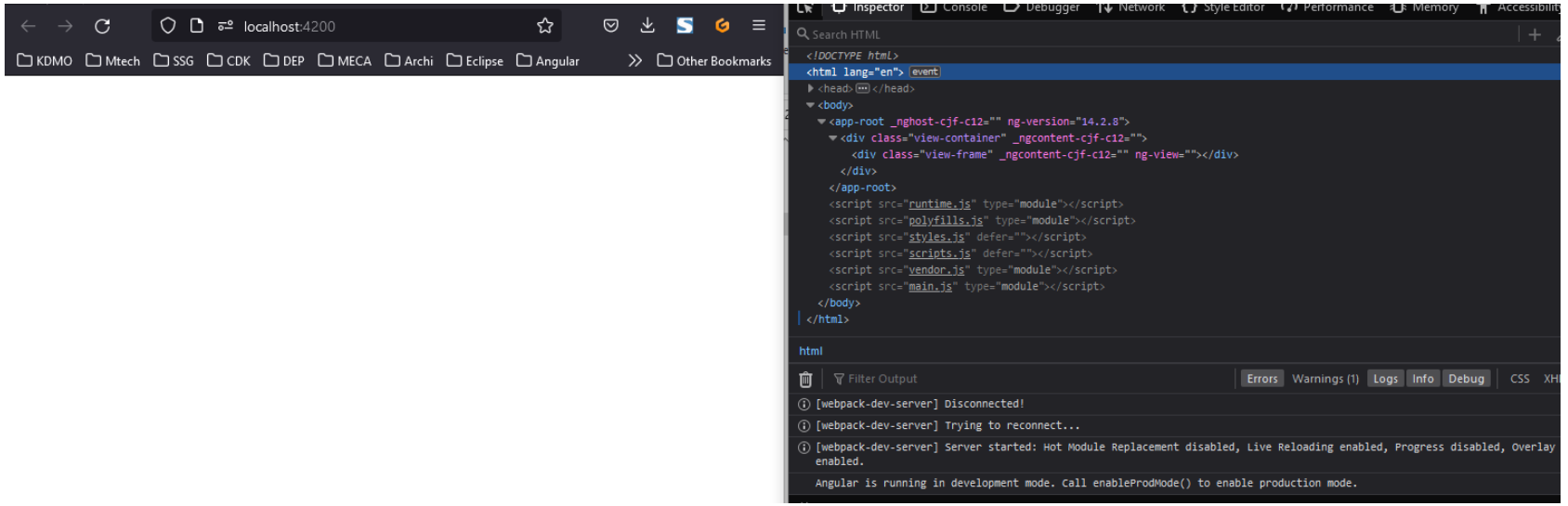
```
if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

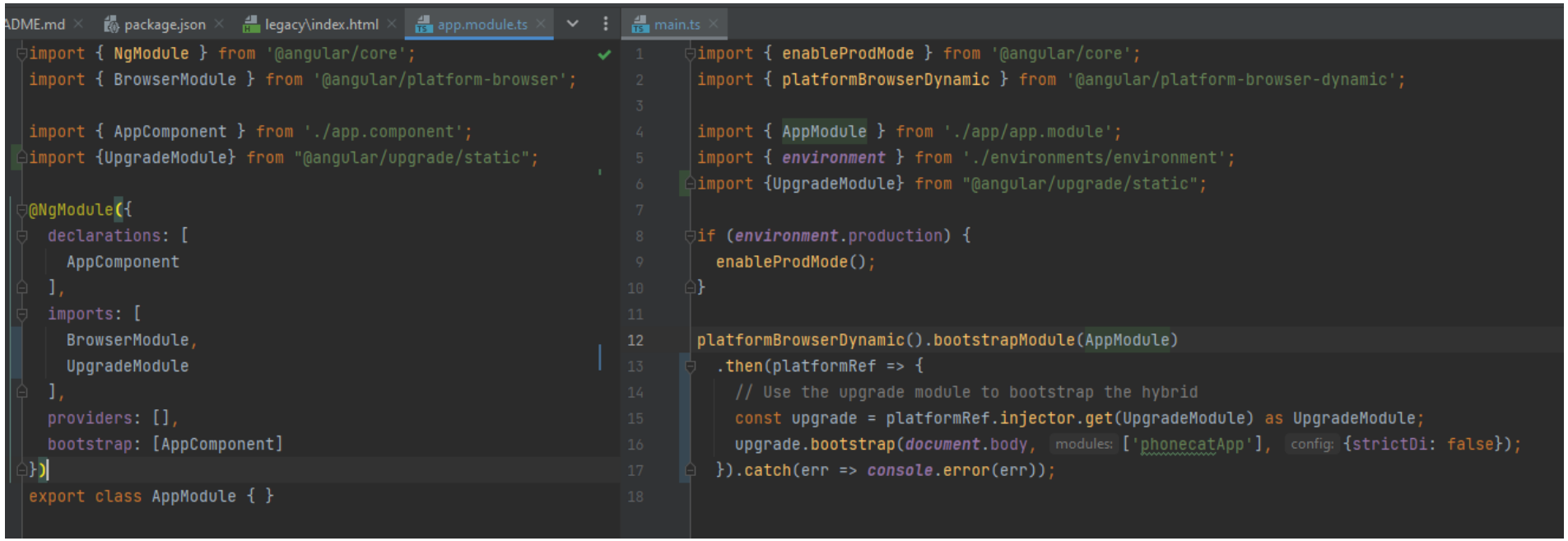
Faisons en sorte que AppComponent soit le point d'entrée du legacy

- Bof bof





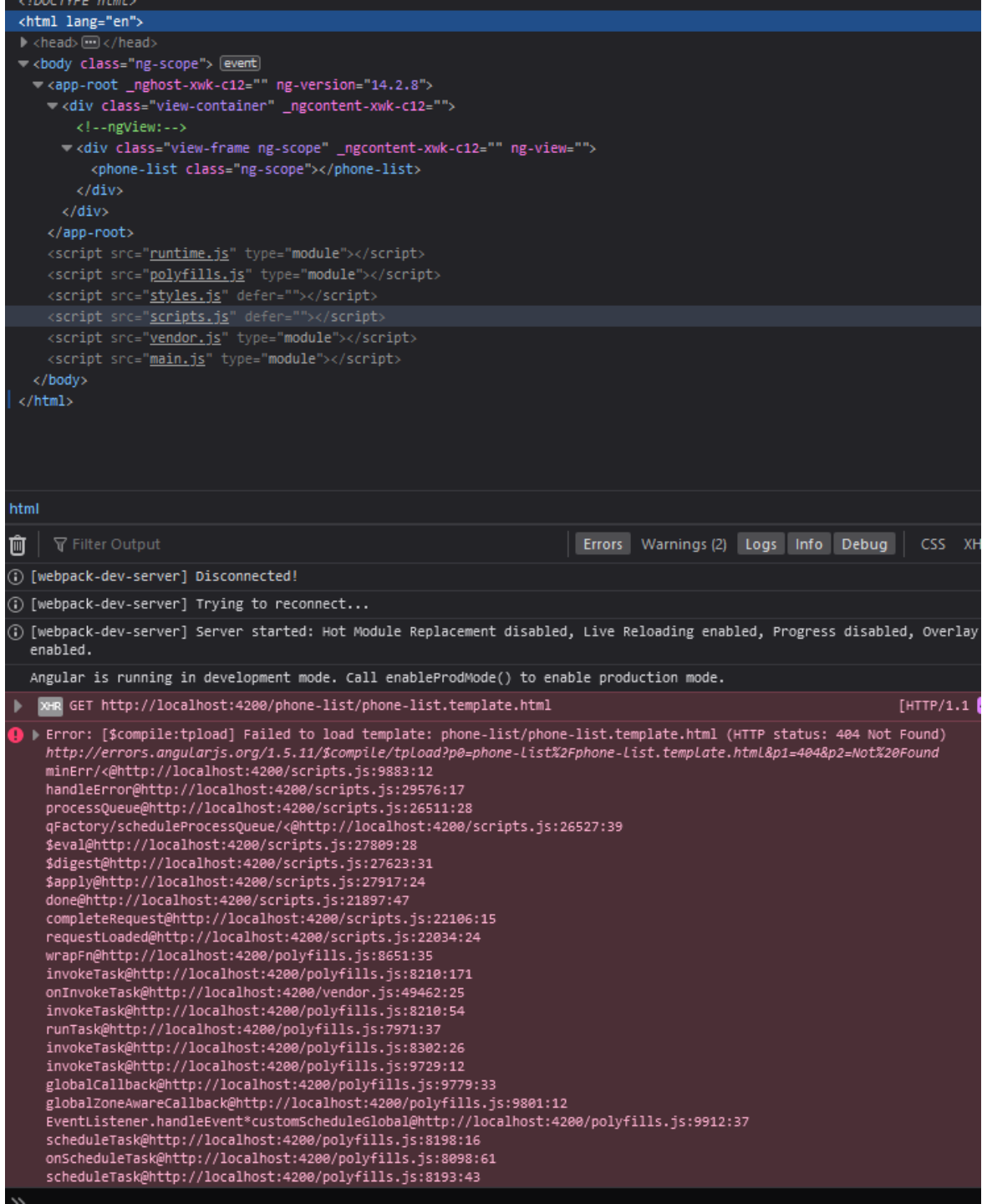
On bootstrap le legacy, pour de vrai



```
ADME.md | package.json | legacy\index.html | app.module.ts

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import {UpgradeModule} from "@angular/upgrade/static";

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    UpgradeModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

```
main.ts

1  import { enableProdMode } from '@angular/core';
2  import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4  import { AppModule } from './app/app.module';
5  import { environment } from './environments/environment';
6  import {UpgradeModule} from "@angular/upgrade/static";
7
8  if (environment.production) {
9    enableProdMode();
10 }
11
12 platformBrowserDynamic().bootstrapModule(AppModule)
13   .then(platformRef => {
14     // Use the upgrade module to bootstrap the hybrid
15     const upgrade = platformRef.injector.get(UpgradeModule) as UpgradeModule;
16     upgrade.bootstrap(document.body, modules: ['phonecatApp'], config: {strictDi: false});
17   }).catch(err => console.error(err));
18
```

- On commence à avoir des choses angular JS !!
- Mais toujours la page blanche



On a load le js, mais pas les templates

- **Google et stackoverflow is your friend : https://stackoverflow.com/questions/59672988/how-to-load-templates-in-angular-hybrid-application**

Try downgrading **@angular-devkit/build-angular** to version **0.802.2**.

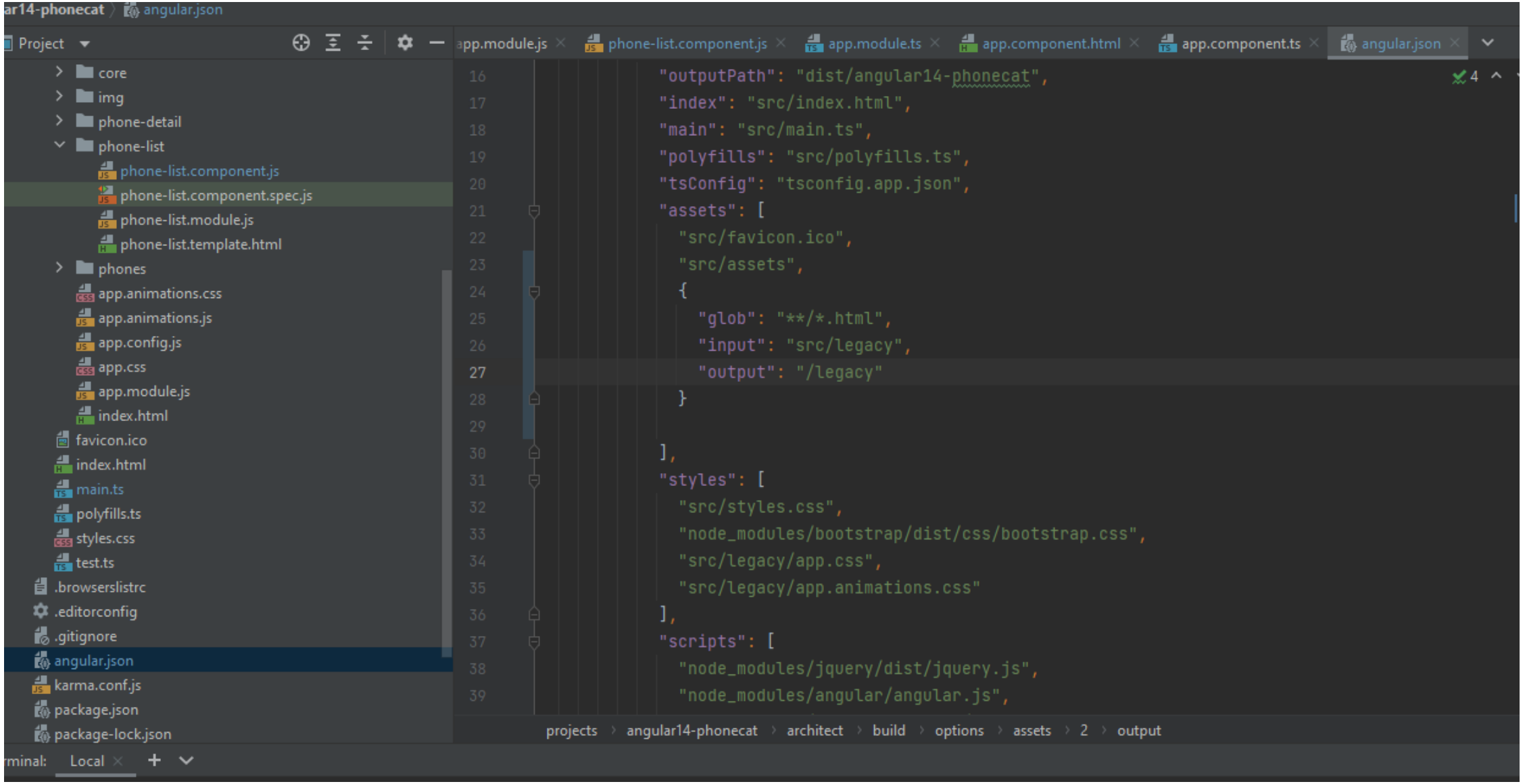
Since **0.803.0** it shows [object Module]

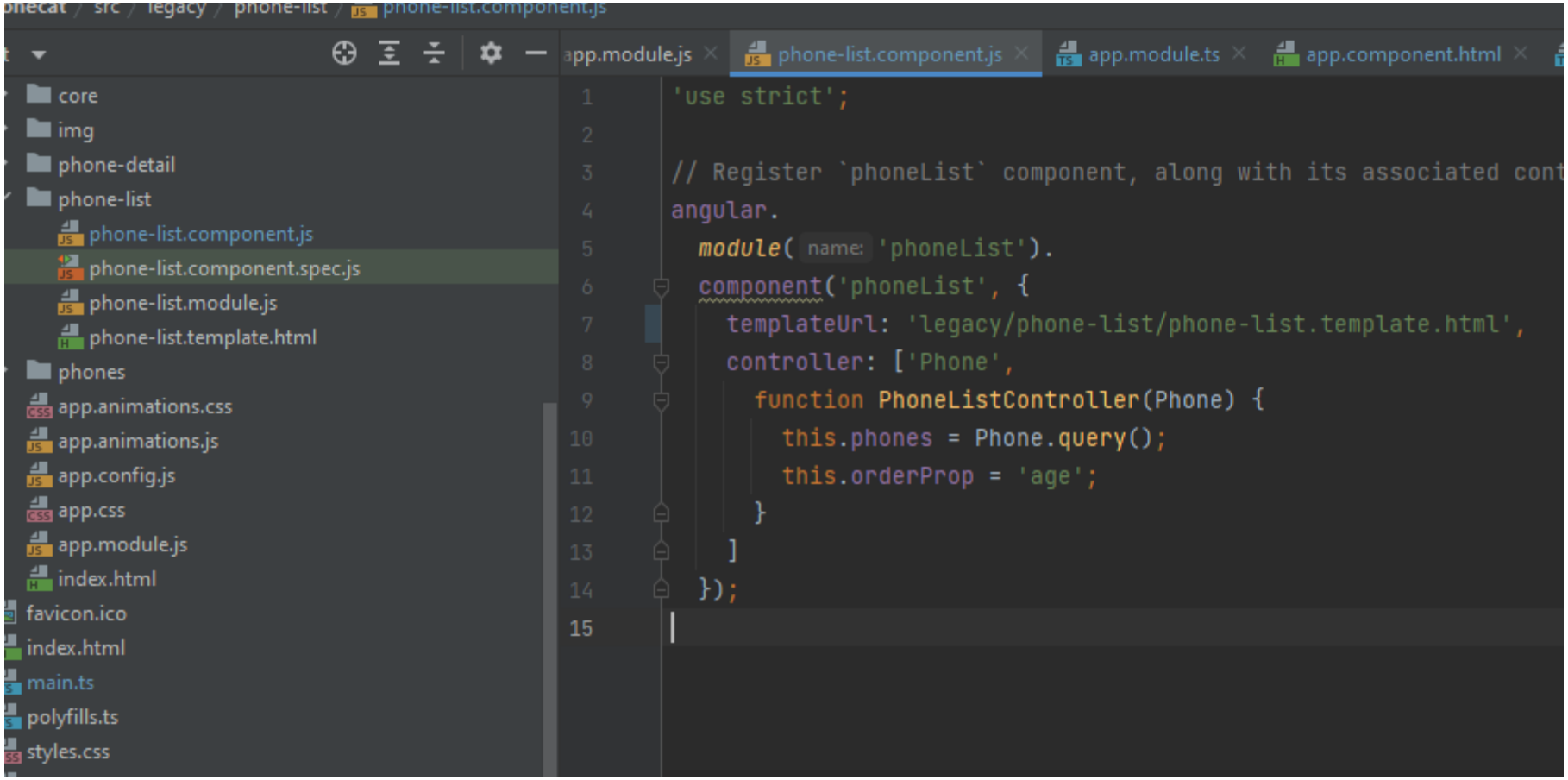
Or you can try completely different approach - instead of using **template: require()** you can use **templateUrl** but then you have to copy html files to destination folder.

To do that, you have to change angular.json and set assets

```
{
    "glob": "**/*.html",
    "input": "apps/lopa/src/app-ajs",
    "output": "/app-ajs"
},
```

And in your component use full path like this:

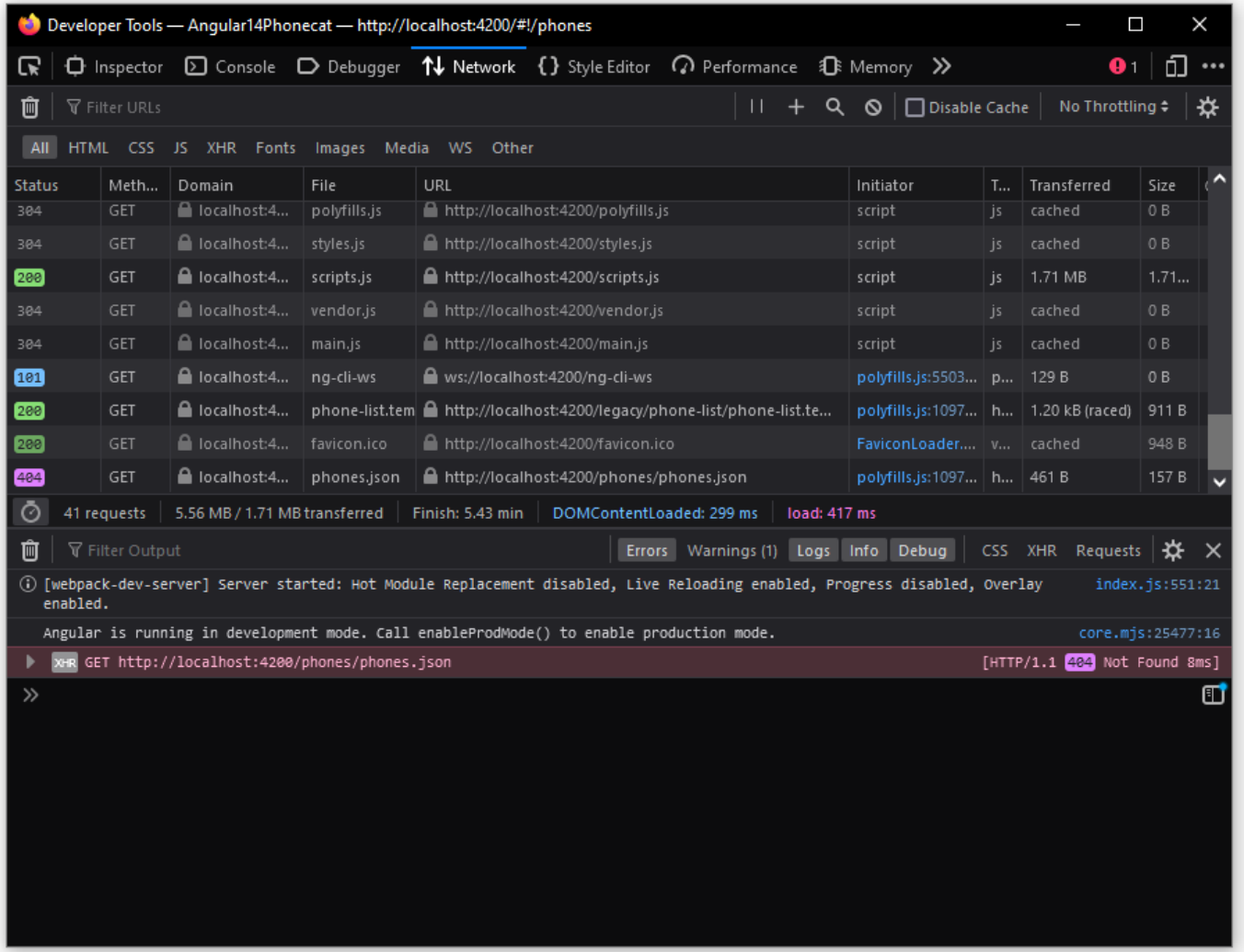



core
img
phone-detail
phone-list
phone-list.component.js
phone-list.component.spec.js
phone-list.module.js
phone-list.template.html
phones
app.animations.css
app.animations.js
app.config.js
app.css
app.module.js
index.html
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
app.module.js
phone-list.component.js
app.module.ts
app.component.html
'use strict';
// Register `phoneList` component, along with its associated cont
angular.
module(name: 'phoneList').
component('phoneList', {
templateUrl: 'legacy/phone-list/phone-list.template.html',
controller: ['Phone',
function PhoneListController(Phone) {
this.phones = Phone.query();
this.orderProp = 'age';
}
]
});

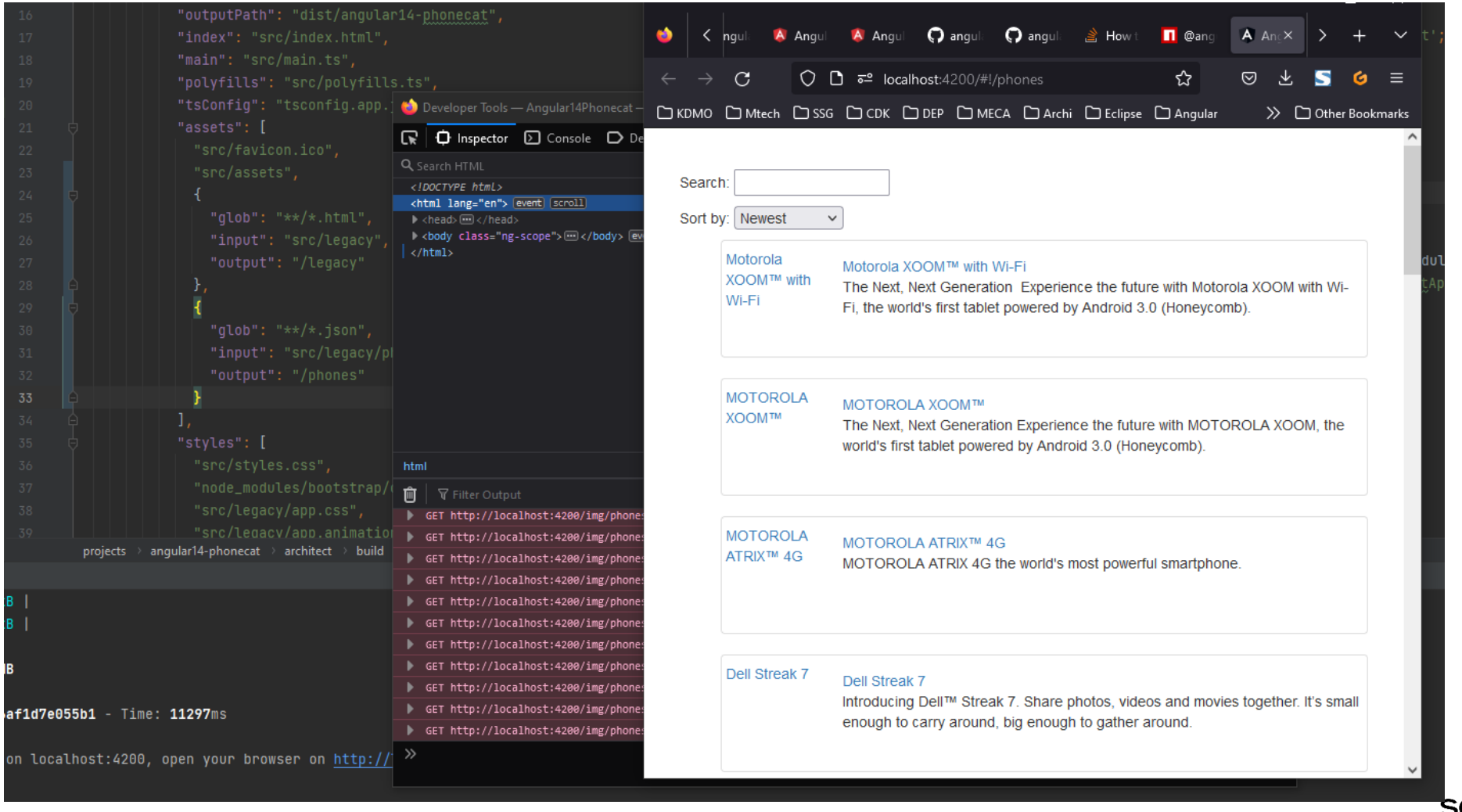
La page n’est plus blanche !!

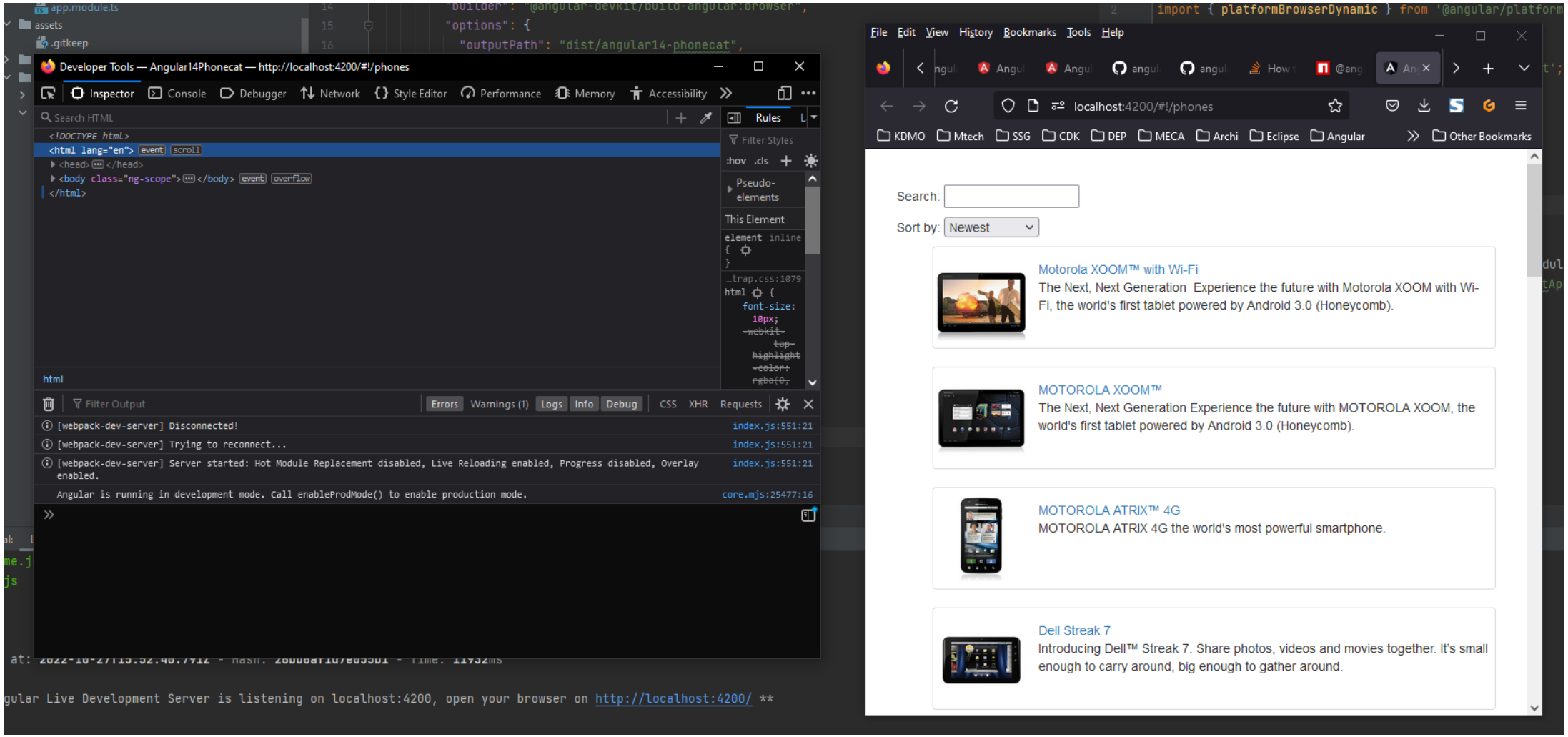
Search:

Sort by: Newest

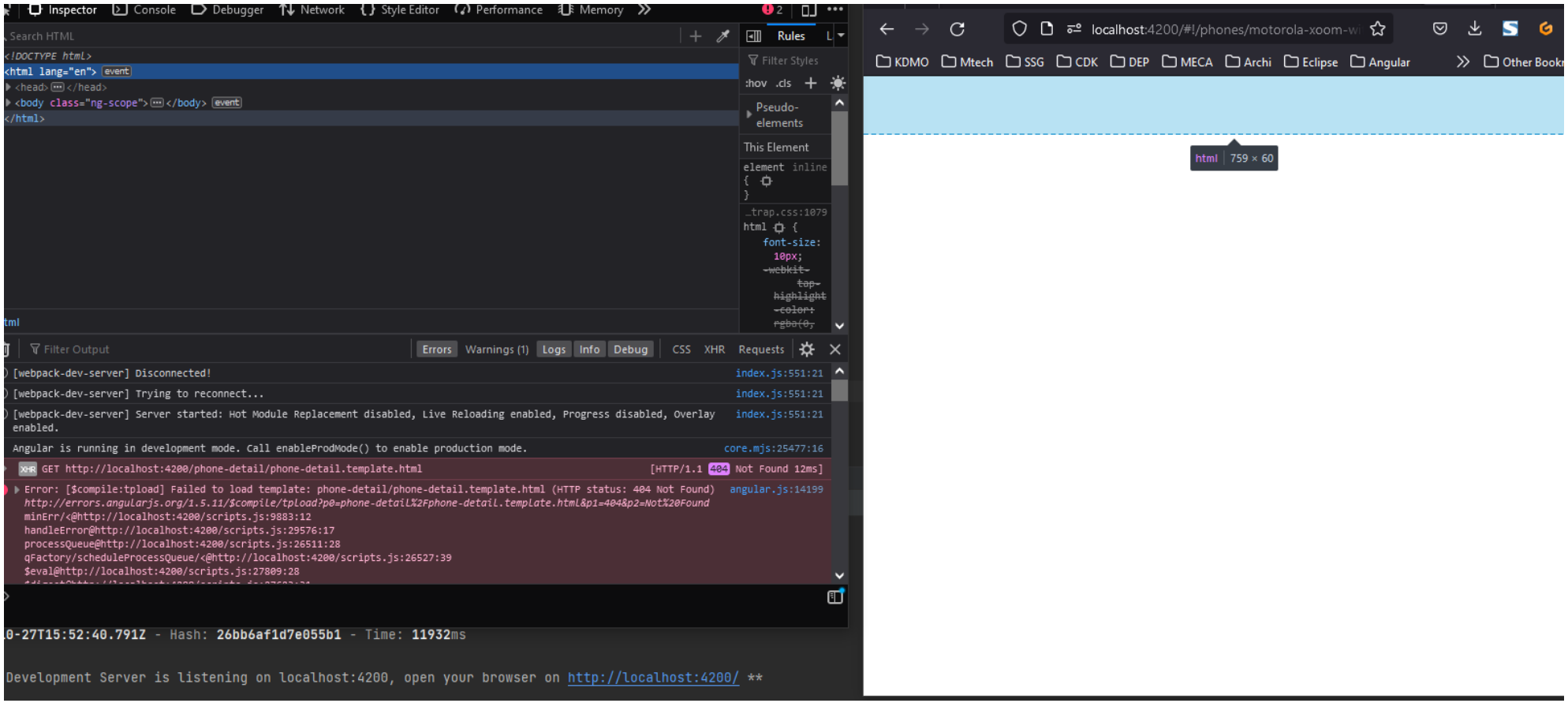


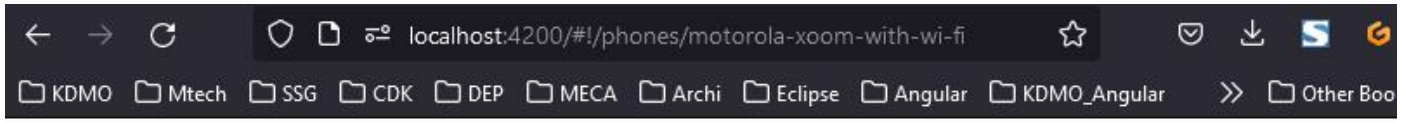
Ajout des assets manquants





Fix template url et relance le serveur



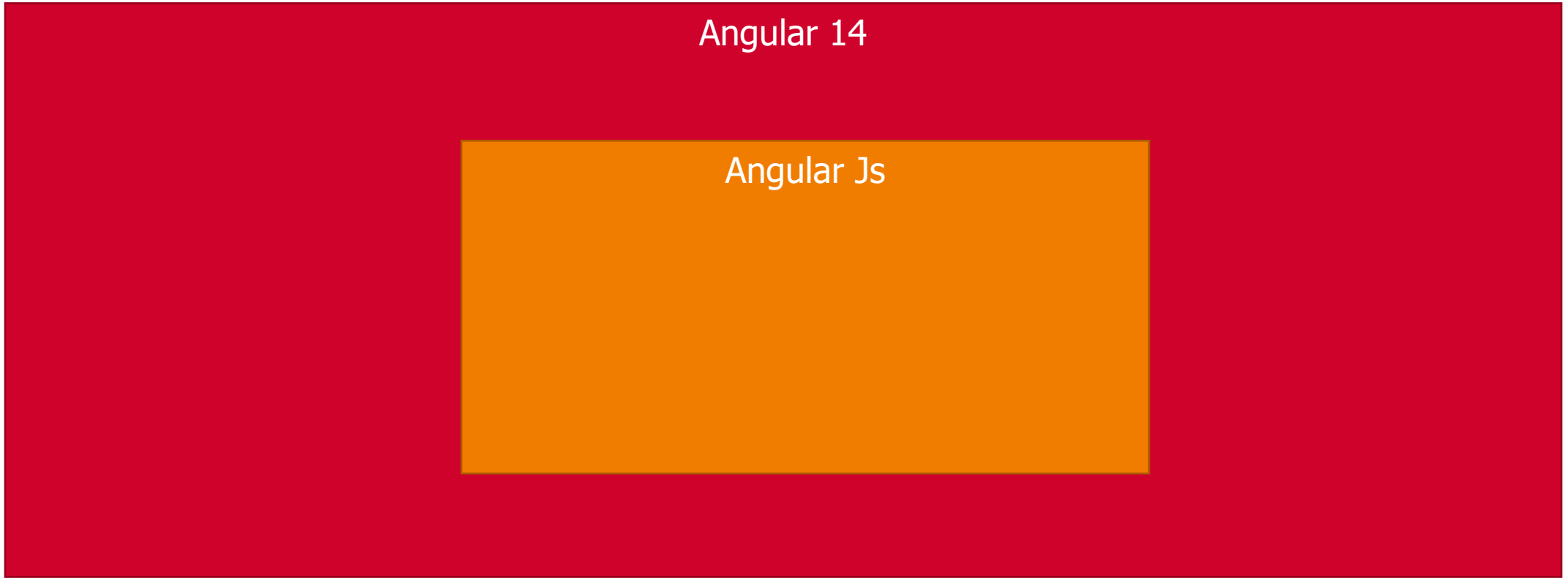


Motorola XOOM™ with Wi-Fi

Motorola XOOM with Wi-Fi has a super-powerful dual-core processor and Android™ 3.0 (Honeycomb) — th Android platform designed specifically for tablets. Witl its 10.1-inch HD widescreen display, you'll enjoy HD video in a thin, light, powerful and upgradeable tablet.

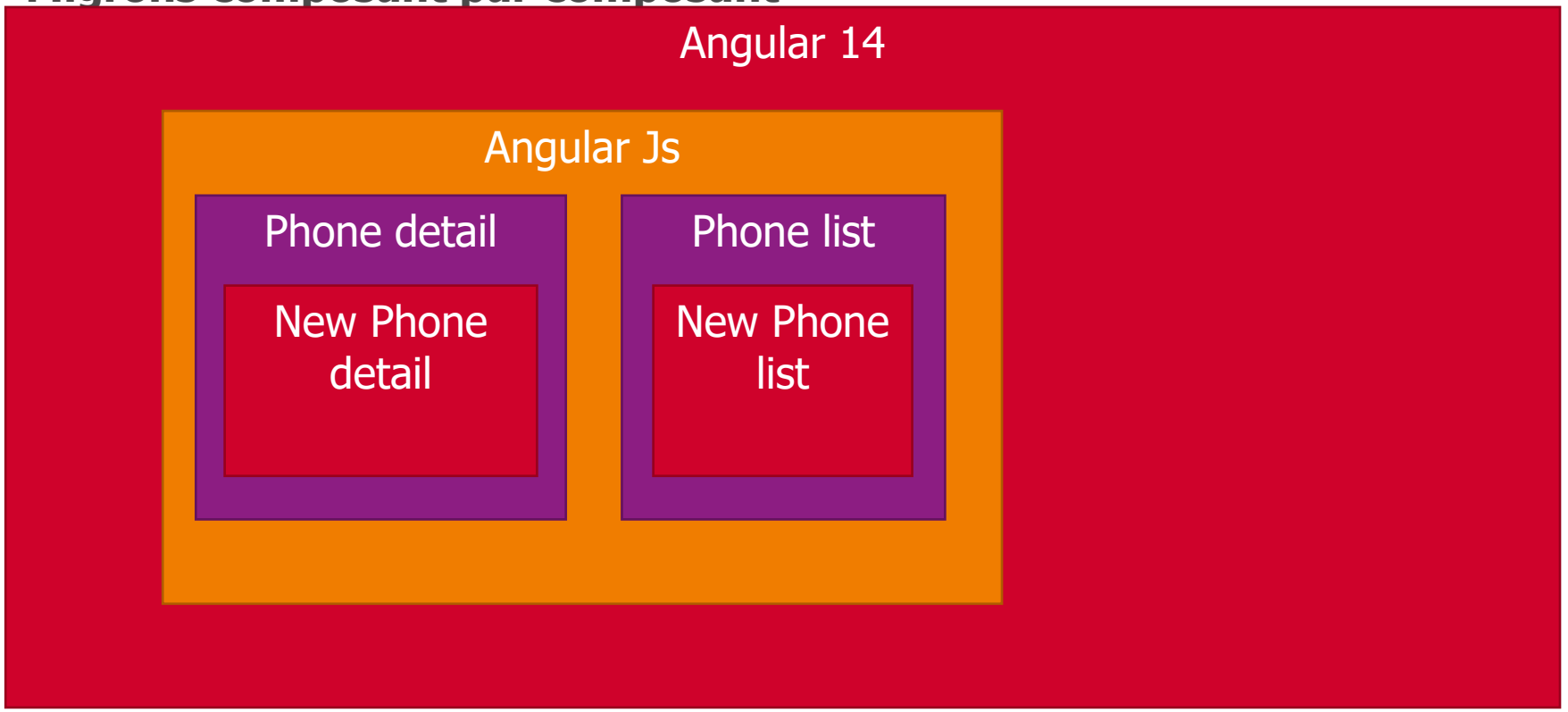
And voila !

- **Angular 14 boostrap de l'angular JS**

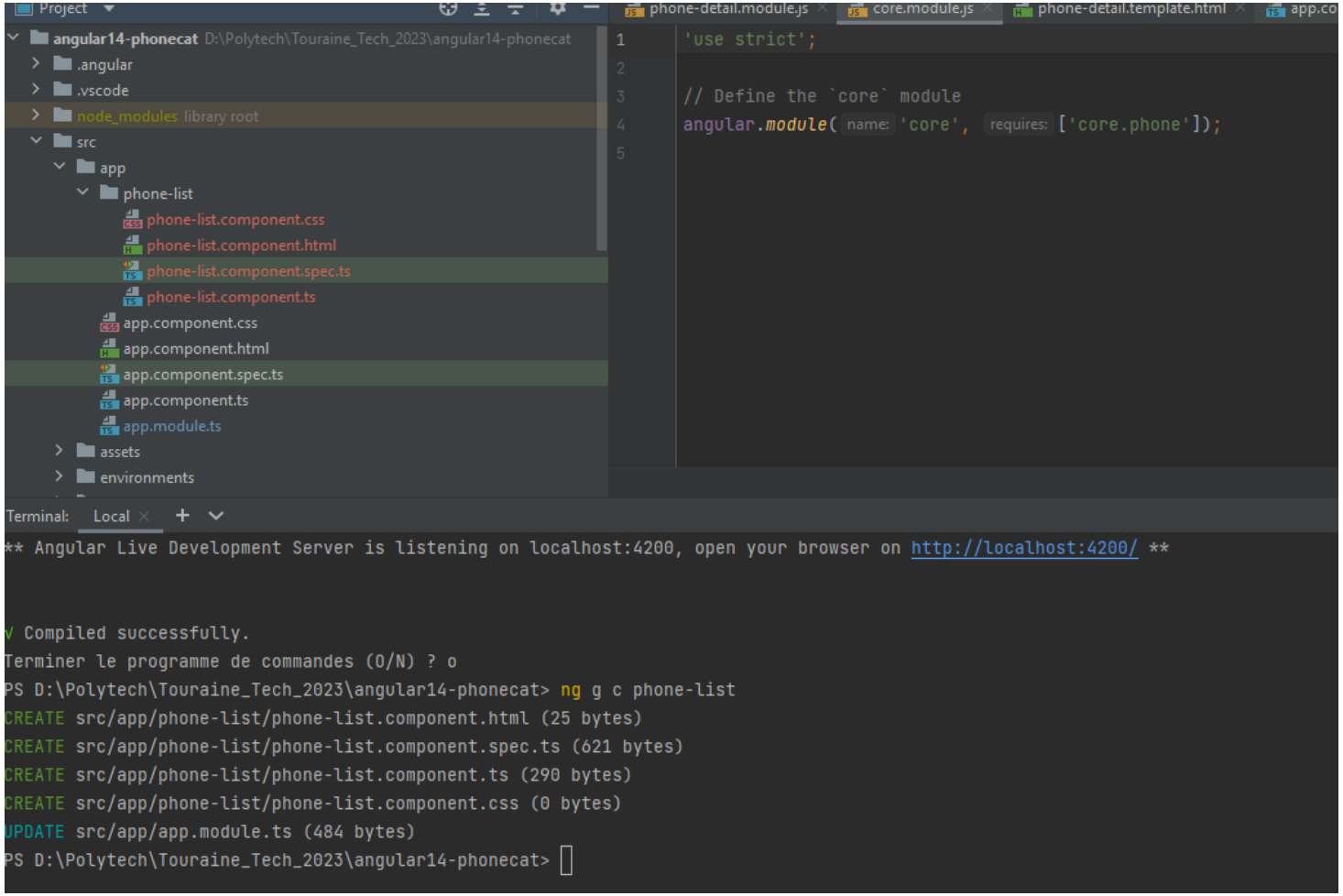


La Suite

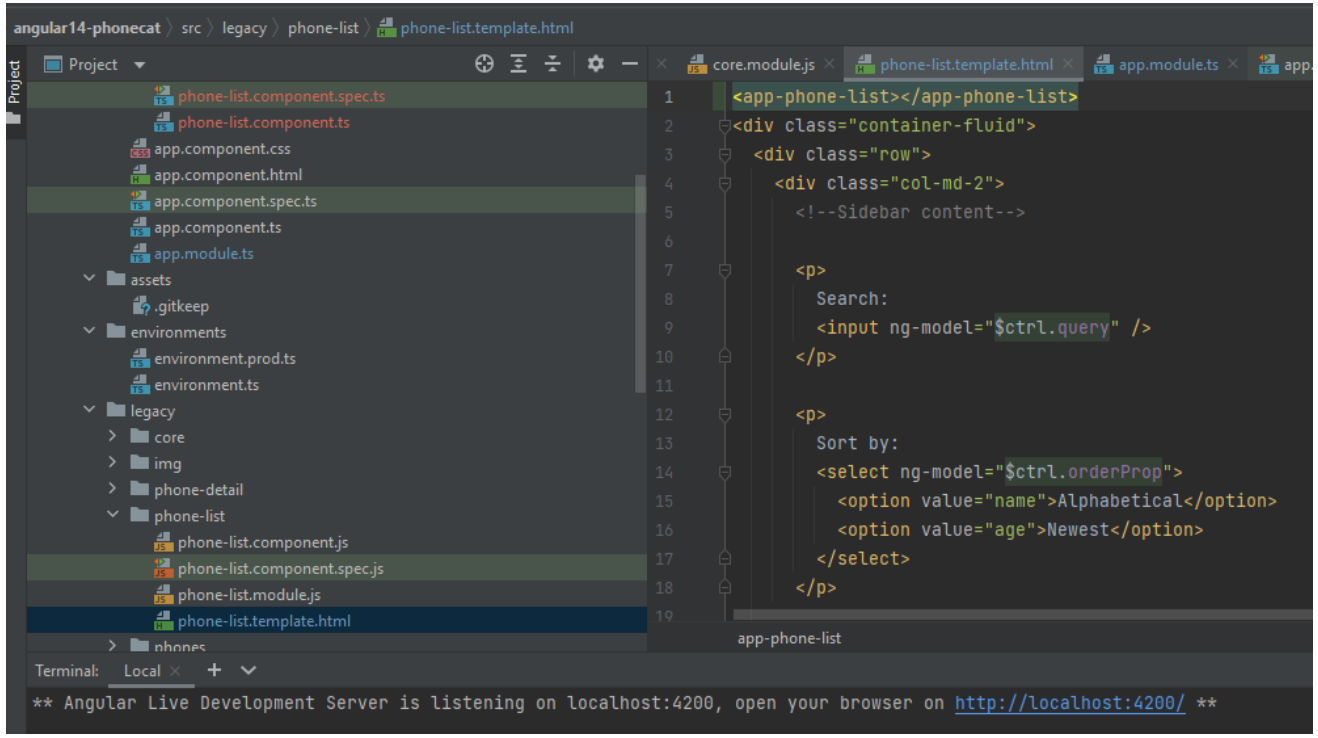
- **Migrons composant par composant**



Nouveau composant phone-list

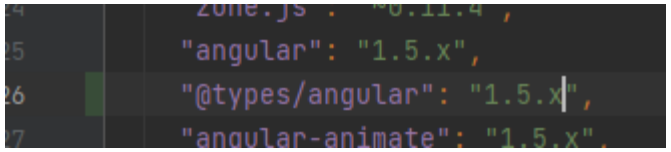


Ajout du nouveau composant dans le html du legacy



On « downgrade » le composant Angular 14

- **La doc dit qu'il faut modifier le module principal du legacy**
  - Et si j'ai envie de faire ça du coté d'Angular 14 ?
  - On a besoin des @types pour le ts



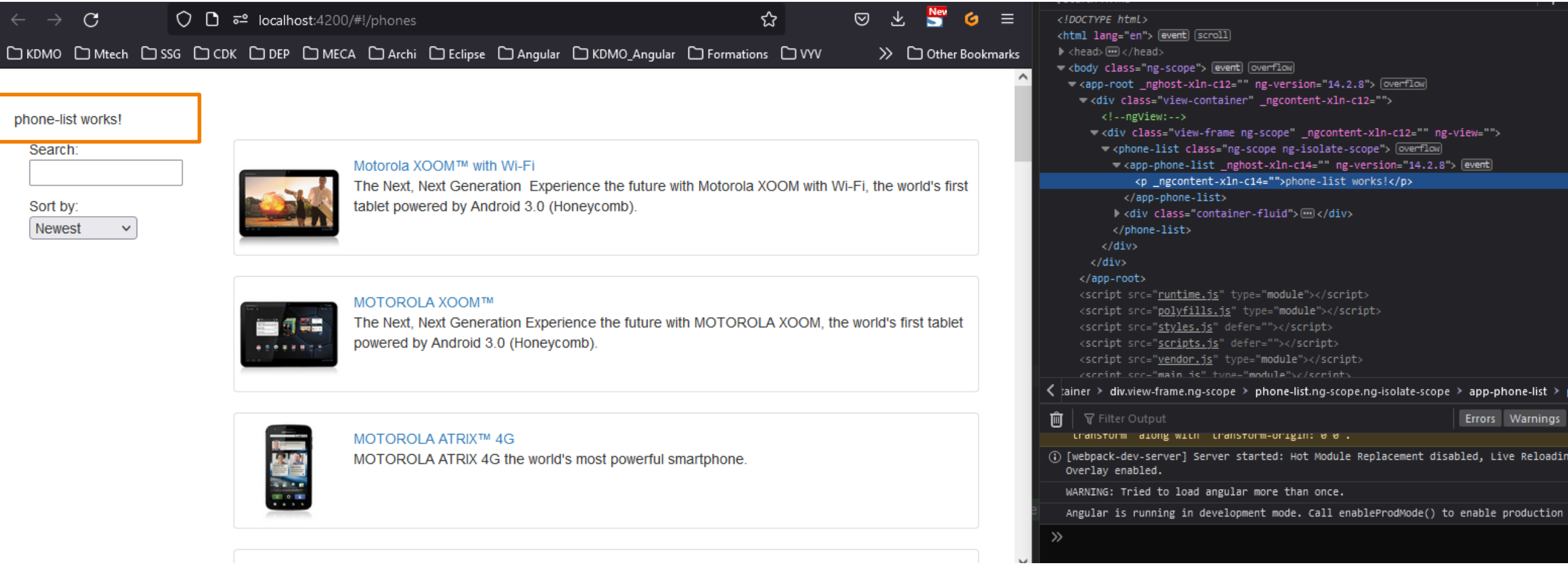
app.module.ts

```ts
import { HeroDetailComponent } from './hero-detail.component';

/* . . . */

import { downgradeComponent } from '@angular/upgrade/static';

angular.module('heroApp', [])
  .directive(
    'heroDetail',
    downgradeComponent({ component: HeroDetailComponent }) as
angular.IDirectiveFactory
  );
```

main.ts | angular.json | package.json | README.md | tsconfig.json | phone-list.component.css | phone-list.component.spec.ts

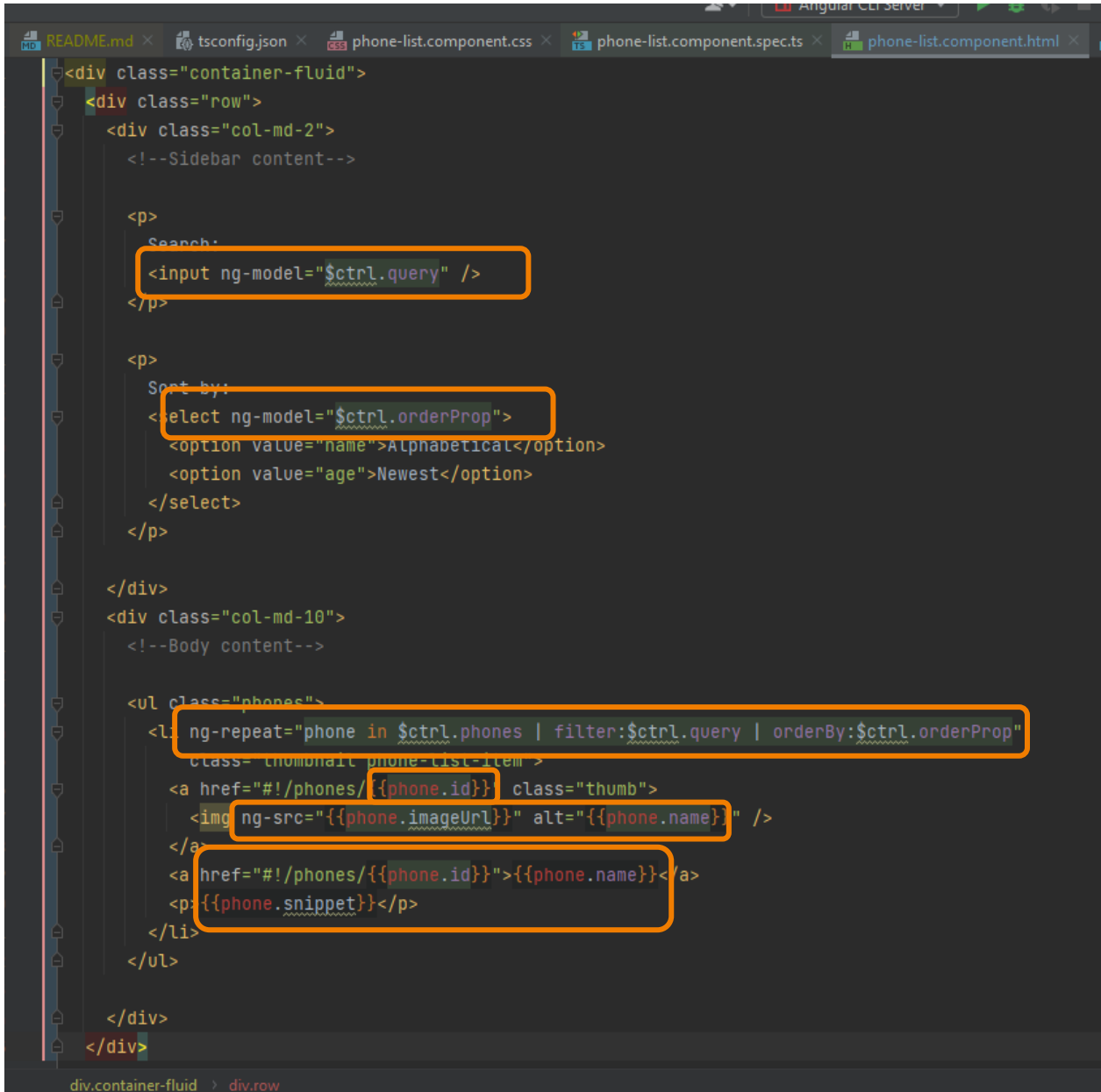
```ts
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app/app.module';
import { environment } from './environments/environment';
import {UpgradeModule, downgradeComponent} from "@angular/upgrade/static";
import {PhoneListComponent} from "./app/phone-list/phone-list.component";
import 'angular';
if (environment.production) {
  enableProdMode();
}
declare const angular: angular.IAngularStatic;
interface DowngradeComponent {
  alias: string;
  ngComponent: any;
}

const ngDowngradeComponents: ReadonlyArray<DowngradeComponent> = [
  // alias must match selector name as CamelCase
  // otherwise angularjs won't load them
  {alias: 'appPhoneList', ngComponent: PhoneListComponent},
];

ngDowngradeComponents.forEach( callbackfn: dgComponent =>
  angular
    .module( name: 'phonecatApp')
    .directive(dgComponent.alias, downgradeComponent( info: {component: dgComponent.ngComponent}) as angular.IDirectiveFactory)
);
```

Composant Angular OK !



Et là c’est à la mimine qu’on migre

- Copier coller le html dans le nouveau composant
- Adapter les directives
- Injection de dépendances/service,etc.



Injection de service legacy dans le composant v14

phone.service.spec.js | phone.module.js | app.module.ts | phone-list.module.js | phone-list.component.spec.js

```
import {Injectable, NgModule} from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import {UpgradeModule} from "@angular/upgrade/static";
import { PhoneListComponent } from './phone-list/phone-list.component';
import {FormsModule} from "@angular/forms";

@Injectable()
export abstract class PhoneService{
  abstract query():any;
}

@NgModule({
  declarations: [
    AppComponent,
    PhoneListComponent
  ],
  imports: [
    BrowserModule,
    UpgradeModule,
    FormsModule
  ],
  providers: [{
    provide: PhoneService, useFactory:(i: any) => i.get('Phone'), deps: ['$injector']
  }],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

PhoneService

```
@Component({
  selector: 'app-phone-list',
  templateUrl: './phone-list.component.html',
  styleUrls: ['./phone-list.component.css']
})
export class PhoneListComponent implements OnInit {
  query: any;
  orderProp="age";
  phones: {id:string, name:string, imageUrl:string, snippet:string}[] = [];

  constructor(private phoneService:PhoneService) { }

  ngOnInit(): void {
    this.phones = this.phoneService.query();
  }

}
```

Ça fonctionne, plus que les pipes

- **RIP ☹**
- **https://v9.angular.io/guide/pipes#appendix-no-filterpipe-or-orderbypipe**

Appendix: No *FilterPipe* or *OrderByPipe*

Angular doesn't provide pipes for filtering or sorting lists. Developers familiar with AngularJS know these as `filter` and `orderBy`. There are no equivalents in Angular.

This isn't an oversight. Angular doesn't offer such pipes because they perform poorly and prevent aggressive minification. Both `filter` and `orderBy` require parameters that reference object properties. Earlier in this page, you learned that such pipes must be impure and that Angular calls impure pipes in almost every change-detection cycle.

Filtering and especially sorting are expensive operations. The user experience can degrade severely for even moderate-sized lists when Angular calls these pipe methods many times per second. `filter` and `orderBy` have often been abused in AngularJS apps, leading to complaints that Angular itself is slow. That charge is fair in the indirect sense that AngularJS prepared this performance trap by offering `filter` and `orderBy` in the first place.

The minification hazard is also compelling, if less obvious. Imagine a sorting pipe applied to a list of heroes. The list might be sorted by hero `name` and `planet` of origin properties in the following way:

```
<!-- NOT REAL CODE! -->
<div *ngFor="let hero of heroes | orderBy:'name,planet'"></div>
```

You identify the sort fields by text strings, expecting the pipe to reference a property value by indexing (such as `hero['name']`). Unfortunately, aggressive minification manipulates the `Hero` property names so that `Hero.name` and `Hero.planet` become something like `Hero.a` and `Hero.b`. Clearly `hero['name']` doesn't work.

While some may not care to minify this aggressively, the Angular product shouldn't prevent anyone from minifying aggressively. Therefore, the Angular team decided that everything Angular provides will minify safely.

The Angular team and many experienced Angular developers strongly recommend moving filtering and sorting logic into the component itself. The component can expose a `filteredHeroes` or `sortedHeroes` property and take control over when and how often to execute the supporting logic. Any capabilities that you would have put in a pipe and shared across the app can be written in a filtering/sorting service and injected into the component.

If these performance and minification considerations don't apply to you, you can always create your own such pipes (similar to the FlyingHeroesPipe) or find them in the community.

Liens/Sources

- https://angular.io/guide/upgrade
- https://github.com/johnpapa/angular-styleguide/blob/master/a1/i18n/fr-FR.md

04

Faisabilité

EN VRAI

Il s'en passe des choses en 7 ans

- **Sortir le Front de son Monolithe**
  - « juste » un dossier ?
  - Il faut revoir le build ?

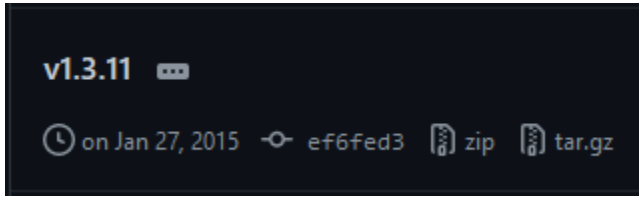
.git	21/02/2022 09:37	Dossier de fichiers
.idea	23/02/2022 08:49	Dossier de fichiers
.mvn	08/02/2022 13:27	Dossier de fichiers
.settings	04/02/2022 11:56	Dossier de fichiers
agirs_batch	07/02/2022 18:18	Dossier de fichiers
agirs_common_business	21/02/2022 10:10	Dossier de fichiers
agirs_common_domain	21/02/2022 10:07	Dossier de fichiers
agirs_common_model	21/02/2022 10:12	Dossier de fichiers
agirs_common_security	07/02/2022 18:18	Dossier de fichiers
agirs_common_tech	21/02/2022 10:06	Dossier de fichiers
agirs_common_test	21/02/2022 10:06	Dossier de fichiers
agirs_common_ws_client	21/02/2022 10:08	Dossier de fichiers
agirs_docasante	21/02/2022 10:10	Dossier de fichiers
agirs_parametrages	21/02/2022 10:08	Dossier de fichiers
agirs_perfs	07/02/2022 18:18	Dossier de fichiers
agirs_sel	07/02/2022 18:18	Dossier de fichiers
agirs_sel_client	21/02/2022 10:09	Dossier de fichiers
agirs_tools	04/02/2022 11:29	Dossier de fichiers
agirs_web	21/02/2022 10:12	Dossier de fichiers
agirs_ws	07/02/2022 18:18	Dossier de fichiers
agirs_ws_audio	07/02/2022 18:18	Dossier de fichiers
agirs_ws_opt	07/02/2022 18:18	Dossier de fichiers
agirs_ws_qualite	07/02/2022 18:18	Dossier de fichiers
agirs_ws_recherche_beneficiaire	07/02/2022 18:18	Dossier de fichiers
agirs_ws_tpg	07/02/2022 18:18	Dossier de fichiers
agirs_ws_tpg_client	21/02/2022 10:09	Dossier de fichiers
agirs_ws_valorisation	07/02/2022 18:18	Dossier de fichiers
.gitignore	04/02/2022 11:28	Document texte
.project	04/02/2022 11:56	Fichier PROJECT
mvnw	02/02/2021 15:24	Fichier
mvnw.cmd	02/02/2021 15:24	Script de comman...
pom.xml	07/02/2022 18:18	Document XML
README.md	04/02/2022 11:28	Fichier MD

Nom	Modifié le	Type	Taille
.idea	18/02/2022 10:26	Dossier de fichiers	
.jhipster	04/02/2022 11:29	Dossier de fichiers	
.settings	08/02/2022 13:25	Dossier de fichiers	
.tmp	08/02/2022 11:32	Dossier de fichiers	
node_modules	21/02/2022 10:13	Dossier de fichiers	
src	04/02/2022 11:29	Dossier de fichiers	
target	21/02/2022 10:13	Dossier de fichiers	
.bowerrc	07/02/2022 18:18	Fichier BOWERRC	1 Ko
.classpath	08/02/2022 13:22	Fichier CLASSPATH	3 Ko
.factorypath	08/02/2022 13:22	Fichier FACTORYP...	37 Ko
.gitignore	04/02/2022 11:29	Document texte	1 Ko
.project	08/02/2022 13:22	Fichier PROJECT	2 Ko
.yo-rc.json	04/02/2022 11:29	Fichier JSON	1 Ko
bower.json	04/02/2022 11:29	Fichier JSON	2 Ko
Gruntfile.js	07/02/2022 18:18	Fichier de JavaScript	17 Ko
npm-shrinkwrap.json	21/02/2022 10:13	Fichier JSON	432 Ko
package.json	07/02/2022 18:18	Fichier JSON	2 Ko
pom.xml	07/02/2022 18:18	Document XML	19 Ko

Il s'en passe des choses en 7 ans

Et chacun y met du sien

- **Beaucoup de dépendances**
  - Parfois dans du bower
  - Parfois en tant qu'asset
- **Du code dupliqué**
- **Les style d'écriture et le pratiques évoluent**
  - Mais pas de façon globale
  - Plusieurs syntaxes pour les modules
  - Pas toujours compatible avec la minification



- **Angular 1.3.11**
  - En 7 ans ça n'a pas bougé
  - https://docs.angularjs.org/guide/migration
  - La roadmap to 1.8 est périlleuse
  - Où sont mes TU pour valider si ça régresse ?

Risques pour la minification

[Style Y090]

- Évitez l'utilisation de la syntaxe raccourcie pour déclaration de dépendances. Ulilisez plutôt une approche compatible avec la minification.

Pourquoi ? : Les paramètres des composants (ex: contrôleur, *factory*, etc.) vont êtres convertis en variables raccourcies. Par exemple, `common` et `dataservice` deviendraient `a` et `b` et ne seraient pas trouvées par Angular.

```
/* à éviter - non compatible avec la minification */
angular
    .module('app')
    .controller('Dashboard', Dashboard);

function Dashboard(common, dataservice) {
}
```

Ce code pourrait produire des variables raccourcies après minification et provoquer des erreurs à l'exécution.

```
/* à éviter - non compatible avec la minification */
angular.module('app').controller('Dashboard', d);function d(a, b) { }
```

Il s'en passe des choses en 7 ans

- **Trop simple à builder !?**

```
grunt.registerTask('build', [
    'regex-check',
    'clean:dist',
    'wiredep:app',
    'includeSource',
    'ngconstant:prod',
    'useminPrepare',
    'ngtemplates',
    'concurrent:dist',
    'concat',
    'copy:dist',
    'ngAnnotate',
    'cssmin',
    'autoprefixer',
    'uglify',
    'rev',
    'usemin',
    'htmlmin'
]);
```

- **C'est quoi l'équivalent Webpack ?**
  - Ça build mais j'ai cassé la minification
  - On a besoin de ngAnnotate

ar.json | tsconfig.app.json | tsconfig.json | webpack-annotate.plugin.js

```
var merge = require('webpack-merge');

exports.default = {
  config: function(cfg) {
    const strategy = merge.strategy({
      'module.rules': 'prepend',
    });

    return strategy (cfg, {
      module: {
        rules: [
          {
            test: /src.*\.js$/,
            use: [{ loader: 'ng-annotate-loader' }],
          }
        ]
      },
    });
  }
}
```

```
PS D:\CBP\agirs-front> ng build --plugin ~webpack-annotate.plugin.js
```

05

Retour d’expérience

Bilan après 8 mois

- **L'avancement ?**
  - 4 mise en production depuis le démarrage du projet
  - Une approche itérative qui fonctionne bien et qui valide le framework hybride
  - Une automatisation des tests de non régression afin de sécurisé le delivery
- **Les difficultés ?**
  - Harmonisation de l'ensemble des composants
- **Les opportunités ?**
  - Amélioration UX/UI