leanovate
beyond product

Upgrade a legacy app to Angular w/ CLI

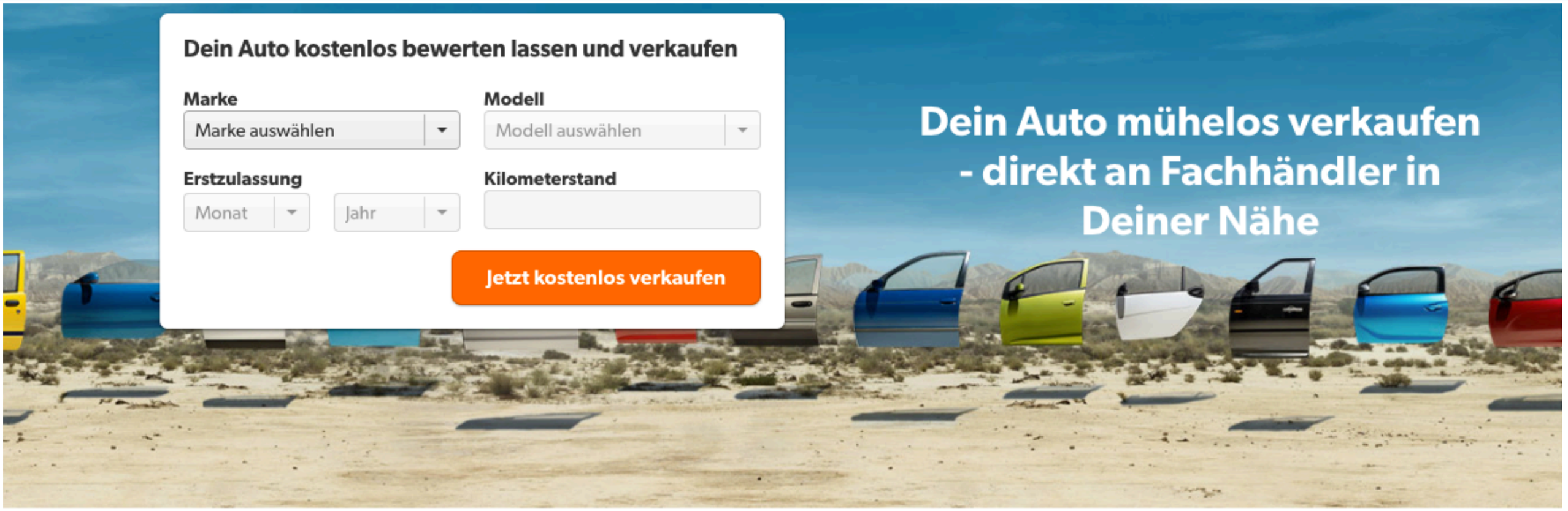
leanovate

beyond product

@devWithDog

The project

leanovate
beyond product



So einfach kann Dein Autoverkauf sein

Kostenlose Preisbewertung

Einfach Autodaten eintragen und kostenlose Preisbewertung für Dein Auto erhalten.

Autofachhändler finden

Vereinbare direkt einen Termin mit einem geprüften Autofachhändler in Deiner Nähe.

Müheloser Verkauf

Kein Verkaufszwang - bei Einigung übernehmen wir die Abmeldung kostenlos.

The project

leanovate
beyond product

leanovate
beyond product

Reality eats plans for breakfast. Be prepared.

www.leanovate.de

leanovate
beyond product

Upgrade to Angular w/ CLI

leanovate
beyond product

angular.io

- Follow the Angular Style Guide
- Using a Module Loader
- Migrating to TypeScript
- Using Component Directives

Upgrade to Angular w/ CLI

Upgrade to Angular w/ CLI

leanovate
beyond product

vsavkin.com: 5 Steps to Angular 2

```
import {NgModule, Component} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {UpgradeModule} from '@angular/upgrade/static';

@Component({
  selector: 'root-cmp',
  template: `
    <div class="ng-view"></div>
  `,
})
export class RootCmp {}

@NgModule({
  imports: [
    BrowserModule,
    UpgradeModule,
  ],

  bootstrap: [RootCmp],
  declarations: [RootCmp]

})

export class Ng2AppModule {

  constructor(public upgrade: UpgradeModule){}

}
```

vsavkin.com: 5 Steps to Angular 2

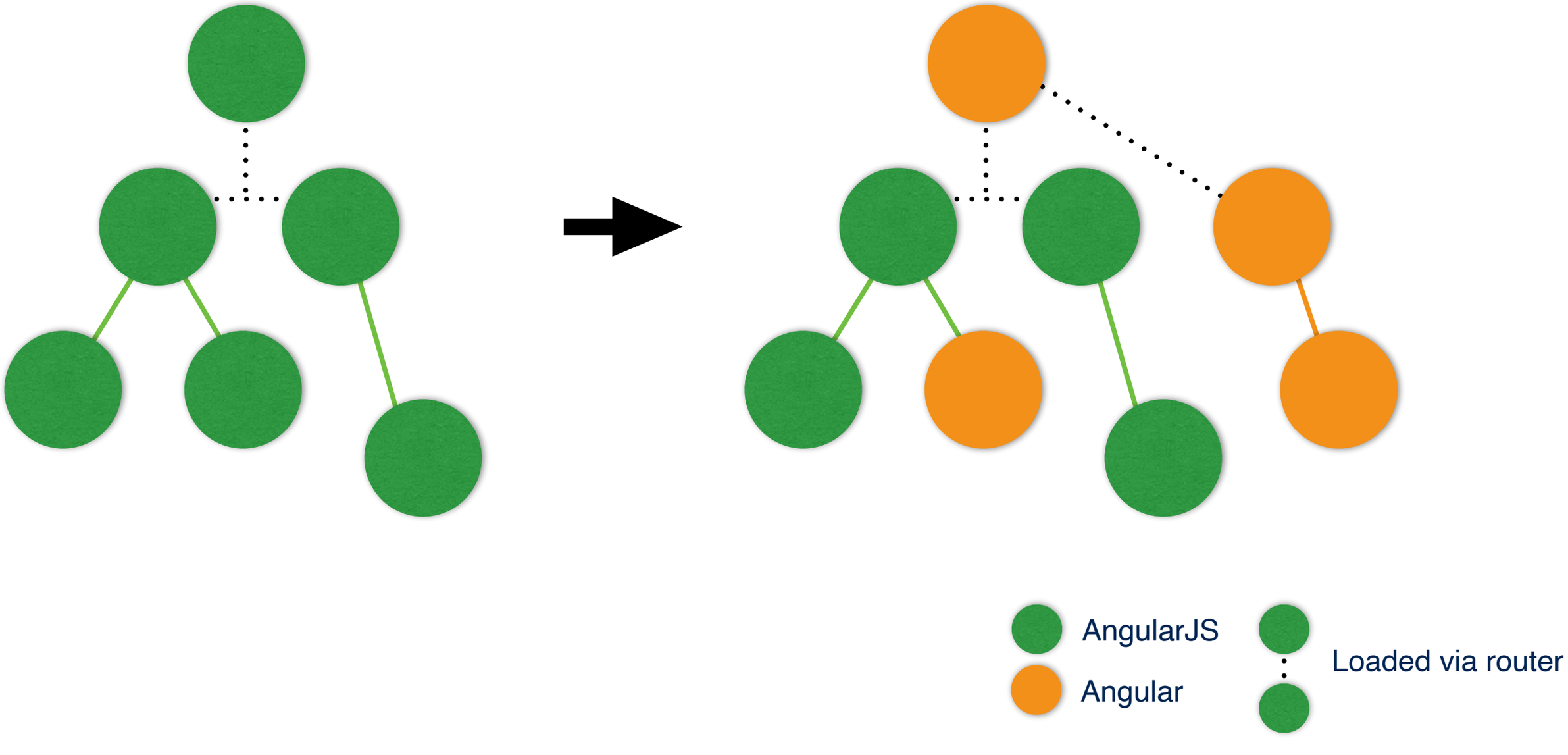
Now, assuming your Angular 1 root module looks something like this:

```
import * as angular from 'angular'

import 'angular-route'

import {MessagesModule} from './messages';

import {MenuModule} from './menu';

import {SettingsModule} from './menu';

export const Ng1AppModule = angular.module('Ng1AppModule‘,[
                                                'ngRoute',
                                                SettingsModule.name,
                                                MessagesModule.name,
                                                MenuModule.name]);
```

Upgrade to Angular w/ CLI

leanovate
beyond product

The Upgrade Shell Strategy



inspired by Victor Savkin

failing forward

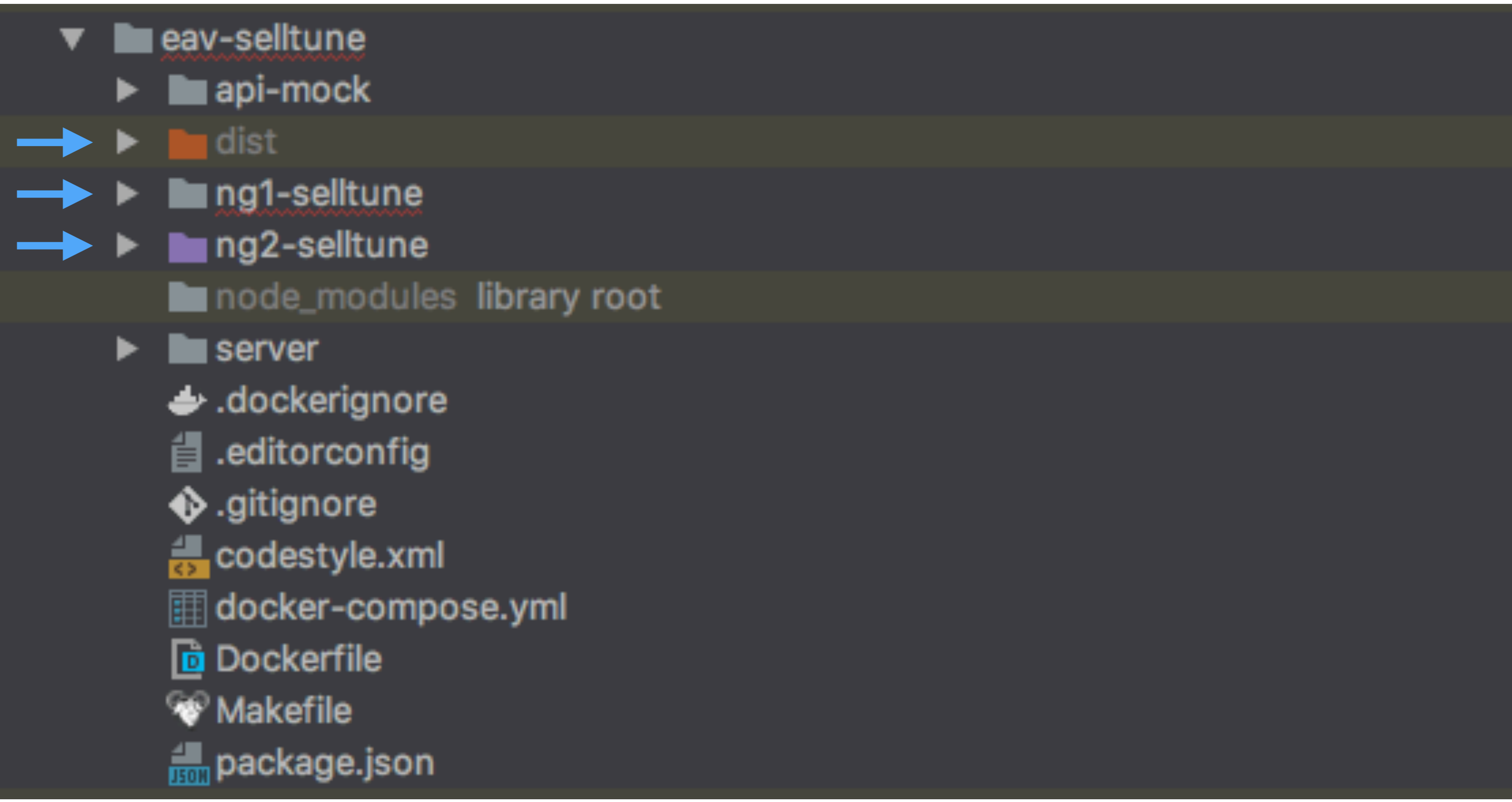
www.leanovate.de

leanovate
beyond product

The solution

leanovate
beyond product

Directory structure



The solution

leanovate
beyond product

Legacy app

```
module.exports = function (grunt) {
    grunt.initConfig({
        pkg         : grunt.file.readJSON('package.json'),
        dist        : '../ng2-selltune/src/dist-ng1',
```

The solution

leanovate
beyond product

Angular CLI

```
"apps": [
    {
        "root": "src",
        "outDir": "dist",
        "assets": [
            "assets",
            "assets/fonts",
            "assets/icons",
            "assets/images",
            "favicon.ico",
            "dist-ng1/icons/font"
        ],
        "index": "index.html",
        "main": "main.ts",
        "polyfills": "polyfills.ts",
        "test": "test.ts",
        "tsconfig": "tsconfig.app.json",
        "testTsconfig": "tsconfig.spec.json",
        "prefix": "mobps",
        "styles": [
            "dist-ng1/icons/css/fontello.css",
            "dist-ng1/app_all.css",
            "scss/styles.scss"
        ],
        "scripts": [
            "dist-ng1/vendor_all.min.js",
            "dist-ng1/SellTune.min.js"
        ],
```

The solution

leanovate
beyond product

Bootstrapping

```
// bootstrap Angular, then bootstrap legacy app
platformBrowserDynamic().bootstrapModule(AppModule).then(platformRef => {
    (<any>platformRef.instance).upgrade.bootstrap(document.body, ['SellTune']);
});
```

The solution

leanovate
beyond product

Url handling

```
@NgModule({
    imports:    [
        RouterModule.forRoot(routes, {useHash: true})
    ],
    exports:    [
        RouterModule
    ],
    // provide custom UrlHandlingStrategy to separate AngularJs from Angular routes
    providers: [
        {
            provide:  UrlHandlingStrategy,
            useClass: Ng1Ng2UrlHandlingStrategy
        }
    ]
})
export class AppRoutingModule {
}
```

The solution

Url handling

```
export class Ng1Ng2UrlHandlingStrategy implements UrlHandlingStrategy {
    private ng1Urls: string[];

    constructor() {
        this.ng1Urls = [
            // ...
        ];
    }

    shouldProcessUrl(url) {
        url = url.toString();
        return this.ng1Urls.findIndex(
                    ng1Url => new RegExp(`^\/${ng1Url}([|\?|\/](.*)){0,1}$`).test(url)
        ) === -1;
    }

    extract(url) {
        return url;
    }

    merge(url, whole) {
        return url;
    }
}
```

The solution

leanovate
beyond product

Template

```
<!-- app.component.html -->
<mobps-header></mobps-header>
<router-outlet></router-outlet>
<div class="ng-view"></div>
```

The solution

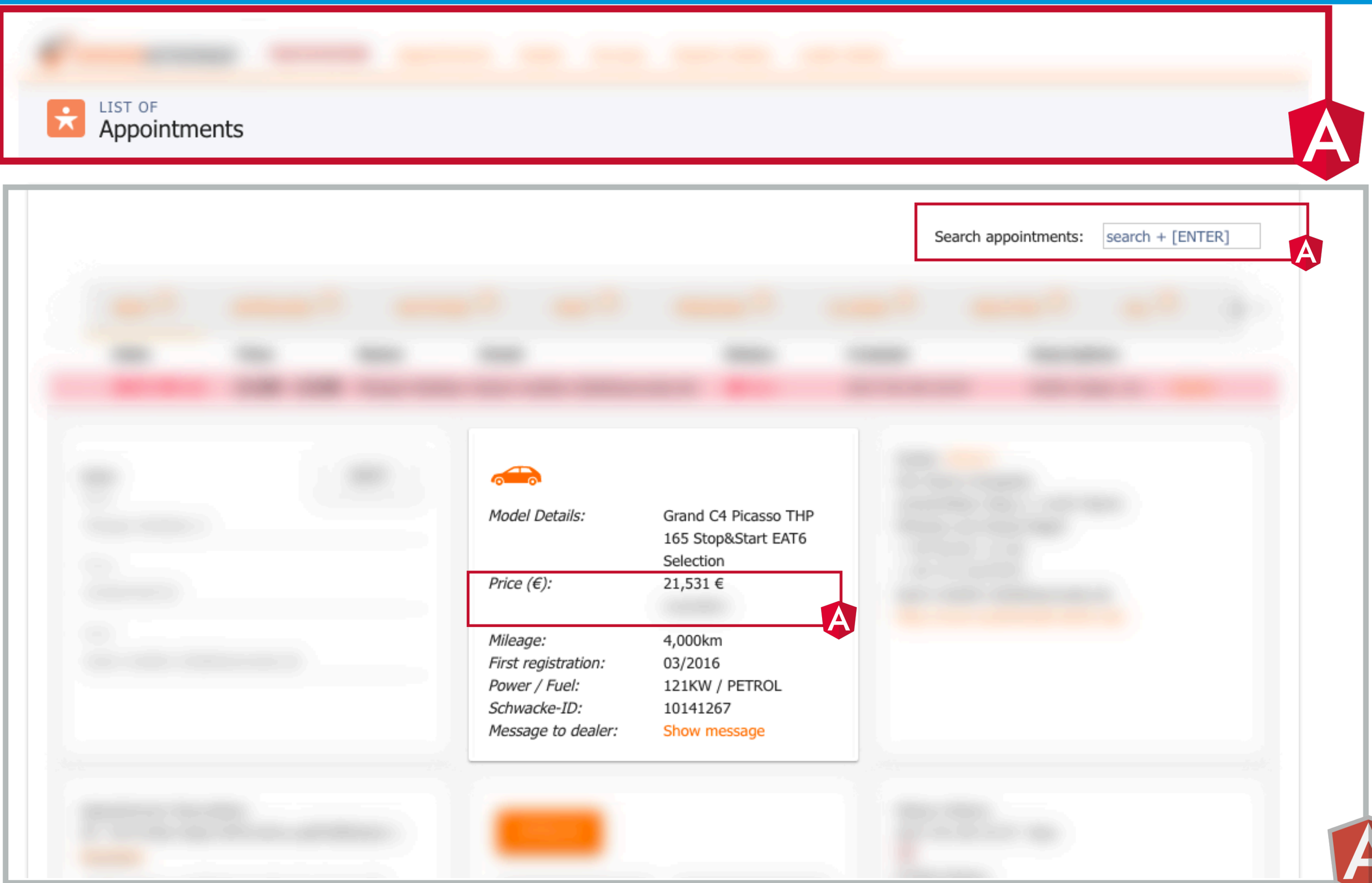
leanovate
beyond product

Downgrading

```
// downgrading services and components for ng1
if (window['angular']) {
    window['angular'].module('SellTune.service')
        .factory(
            'Ng2PageTitleService',
            downgradeInjectable(PageTitleService));

    window['angular'].module('SellTune.service')
        .factory(
            'EnvService',
            downgradeInjectable(EnvService));

    window['angular'].module('SellTune')
        .directive(
            'search',
            downgradeComponent({component: SearchComponent, outputs: ['search']})
        );

    window['angular'].module('SellTune')
        .directive(
            'carPrice',
            downgradeComponent({component: CarPriceComponent, inputs: ['carPrice']})
        );
}
```

The result

leanovate
beyond product



Worth mentioning

leanovate
beyond product

Cross app communication

```
angular.module('SellTune.appointment').controller('AppointmentsNg1Ctrl',
AppointmentCtrl);

function AppointmentCtrl(... ,Ng2PageTitleService, ...) {
    var am = this;
    // ...
    am.$onInit = function() {
        // ...
        Ng2PageTitleService.setTitle({
            title: 'Appointments',
            subTitle: 'List of',
            icon: {
                icon: 'lead', category: 'standard'
            }
        });
    };
    // ...
}
```

Worth mentioning

leanovate
beyond product

Cross app communication

```
// page-title.service.ts
@Injectable()
export class PageTitleService {
    private pageTitleSource = new ReplaySubject<PageTitle>();

    setTitle(pageTitle: PageTitle) {
        this.pageTitleSource.next(pageTitle);
    }

    pageTitle(): Observable<PageTitle> {
        return this.pageTitleSource.asObservable();
    }
}

// header.component.ts
this.pageTitleSubscription = pageTitleService.pageTitle$
    .subscribe((pageTitle: PageTitle) => {
        titleService.setTitle(`${pageTitle.subTitle} ${pageTitle.title}`);
        this.currentNavItem = pageTitle;
        this.cd.markForCheck();
    });
```

Worth mentioning

leanovate
beyond product

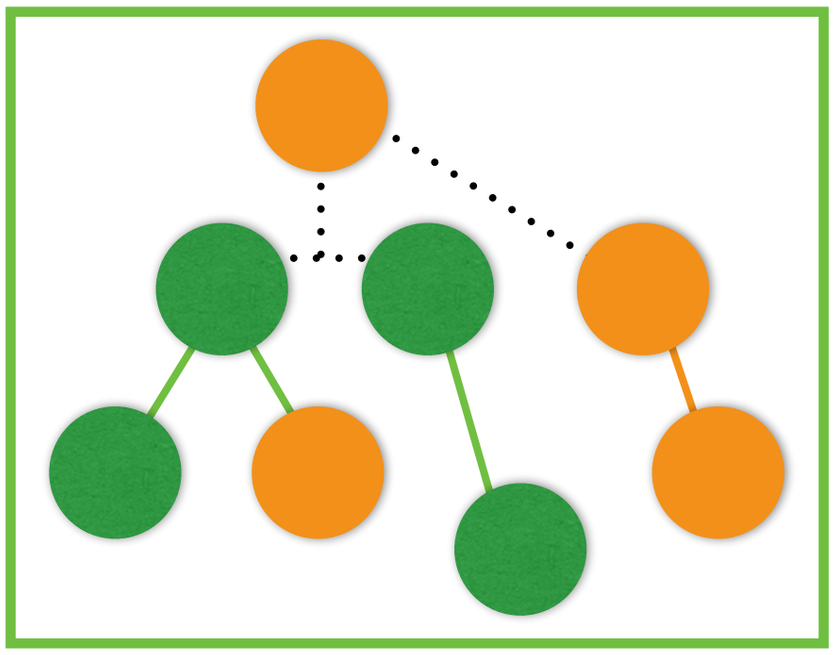
Pitfalls

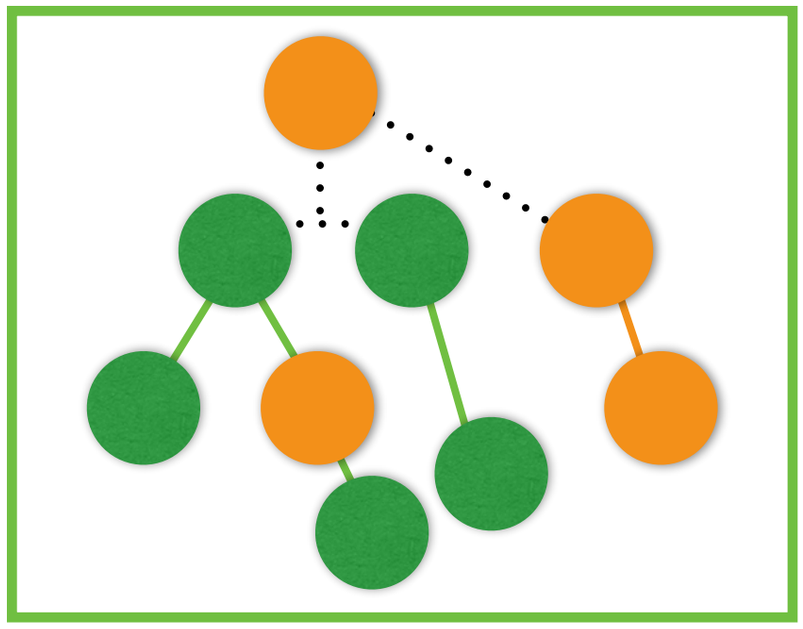
- Only hash urls work out of the box
  - The Angular apps don't empty their router outlets
- ng1 *otherwise* must not be used
- Initial redirect to ng1 route:
  - There has to be a „fake“ Angular route

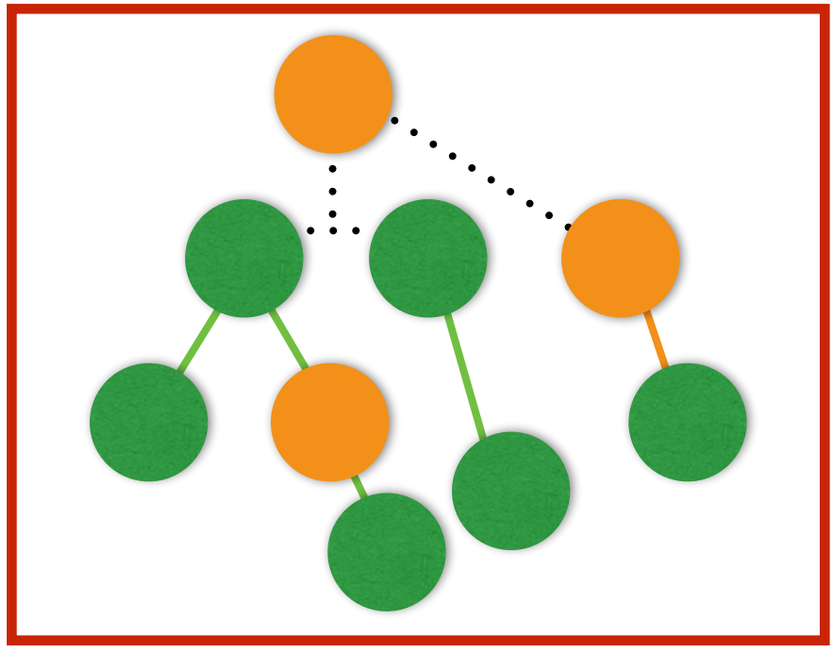
Worth mentioning

leanovate
beyond product

Pitfalls







AngularJS

Angular

leanovate
beyond product

we are hiring

jobs@leanovate

leanovate
beyond product

Questions?