

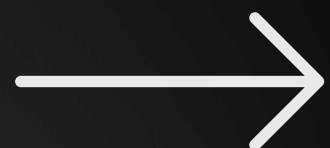
INTRODUCTION AUX STANDALONE COMPONENTS ANGULAR

Une Révolution dans l'Écosystème
Angular



Elodie TURLIER

Angular, IA, agilité &
storytelling : construire des apps
prêtes à survivre à l'apocalypse
digitale



INTRODUCTION

Les standalone components constituent un changement de paradigme majeur dans l'écosystème Angular, introduits avec Angular 14 et stabilisés avec Angular 15.

✔ Simplification de l'architecture Angular

✔ Suppression de la complexité des NgModules

✔ Fonctionnement indépendant des composants

✔ Architecture plus intuitive et moderne

Cette révolution transforme fondamentalement la façon dont nous développons des applications Angular.



Elodie TURLIER

Angular, IA, agilité & storytelling



DÉFINITION

Un standalone component est un composant Angular autonome qui n'a pas besoin d'être déclaré dans un NgModule.

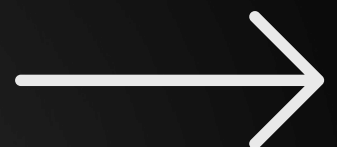
```

@Component({
  selector: 'app-example',
  standalone: true, // n'est plus utile depuis Angular 19 (standalone par défaut)
  template: `
    <div>
      <h1>{{ title }}</h1>
      <p>Un composant standalone autonome</p>
    </div>
  `
})
export class ExampleComponent {
  title = 'Mon Standalone Component';
}
```



Elodie TURLIER

Angular, IA, agilité & storytelling



CARACTÉRISTIQUES

✓ Propriété standalone: true dans le décorateur

par default depuis Angular 19

✓ Gestion des dépendances via imports

✓ Élimination du code boilerplate

✓ Granularité fine des dépendances



Elodie TURLIER

Angular, IA, agilité & storytelling



AVANTAGES

Les standalone components apportent des améliorations significatives dans tous les aspects du développement Angular.

Simplification de l'Architecture

Élimination des hiérarchies complexes de NgModules

Performance Optimisée

Meilleur tree-shaking et bundles plus petits

Testabilité Améliorée

Tests unitaires isolés et prédictibles

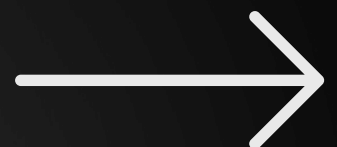
Maintenance Simplifiée

Dépendances localisées et explicites



Elodie TURLIER

Angular, IA, agilité & storytelling



CRÉATION AVEC ANGULAR CLI

Angular CLI offre un support natif pour créer et configurer des standalone components.

Créer un standalone component

ng generate component mon-component --standalone

Nouveau projet standalone

ng new mon-projet --standalone

Configuration globale

```
{
  "projects": {
    "mon-projet": {
      "schematics": {
        "@schematics/angular:component": {
          "standalone": true
        }
      }
    }
  }
}
```



Elodie TURLIER

Angular, IA, agilité & storytelling



MIGRATION DEPUIS 06 LES NGMODULES

La migration vers les standalone components peut s'effectuer progressivement sans disruption majeure.

Commande de migration automatique ()dispo à partir de 15.2 :

ng generate @angular/core:standalone

Étapes de migration :

- 1 - Analyse automatique du code existant
- 2 - Conversion progressive des composants
- 3 - Test de chaque étape de migration
- 4 - Optimisation post-migration

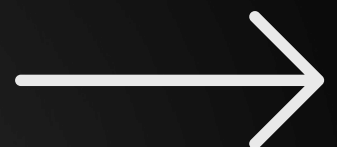
Voir la documentation officielle :

<https://angular.dev/reference/migrations/standalone>



Elodie TURLIER

Angular, IA, agilité & storytelling



ROUTING

Le routing moderne utilise `provideRouter` et permet un lazy loading granulaire.

```
// Bootstrap moderne
bootstrapApplication(MainComponent, {
  providers: [provideRouter(routes)]
});

// Routes avec lazy loading
export const routes: Route[] = [
  {
    path: 'dashboard',
    loadComponent: () => import('./dashboard/dashboard.component')
  },
  {
    path: 'products',
    loadChildren: () => import('./products/products.routes')
  }
];
```

✓ Élimination de `RouterModule.forRoot()`

✓ Fonction `provideRouter` simplifiée

✓ Lazy loading au niveau composant

✓ Contrôle granulaire du chargement



Elodie TURLIER

Angular, IA, agilité & storytelling



BOOTSTRAP D'APPLICATION

Le bootstrap utilise `bootstrapApplication` au lieu des modules traditionnels.

```
// Bootstrap d'application standalone
bootstrapApplication(MainComponent, {
  providers: [
    provideRouter(routes),
    provideHttpClient(),
    // autres providers globaux
  ]
});
```

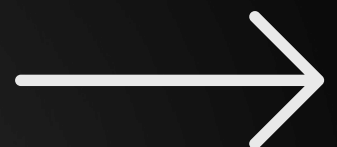
- ✓ Fonction **`bootstrapApplication`**
- ✓ **Élimination** du module racine
- ✓ **Configuration centralisée des providers**
- ✓ Compatible **SSR** et environnements avancés

L'application devient **plus légère** avec une **structure plus claire**.



Elodie TURLIER

Angular, IA, agilité & storytelling



BONNES PRATIQUES

Optimisez vos standalone components avec ces bonnes pratiques essentielles.

Structuration des Imports

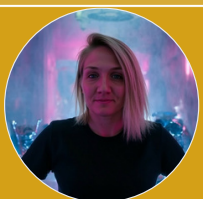
- Importer uniquement les dépendances nécessaires
- Organiser logiquement les imports
- Éviter les imports massifs inutiles

Réutilisabilité

- Créer des composants autonomes
- Favoriser l'architecture orientée composants
- Développer des design systems

Performance

- Utiliser le lazy loading granulaire
- Implémenter OnPush strategy
- Optimiser le préchargement



Elodie TURLIER

Angular, IA, agilité & storytelling



CONCLUSION

10

Les standalone components transforment Angular vers une approche plus moderne et accessible.

Perspectives d'Avenir

- ✓ Architecture moderne et optimisée
- ✓ Écosystème en évolution continue
- ✓ Formation des équipes essentielle
- ✓ Innovation continue promise

Recommandations

→ *Nouveaux projets*

Adoption dès l'initialisation recommandée

→ *Projets existants*

Migration progressive conseillée

→ *Équipes*

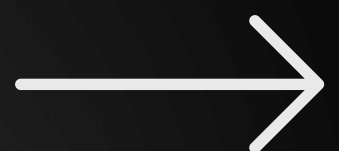
Investissement en formation nécessaire

Les standalone components positionnent Angular comme un framework plus intuitif et moderne.

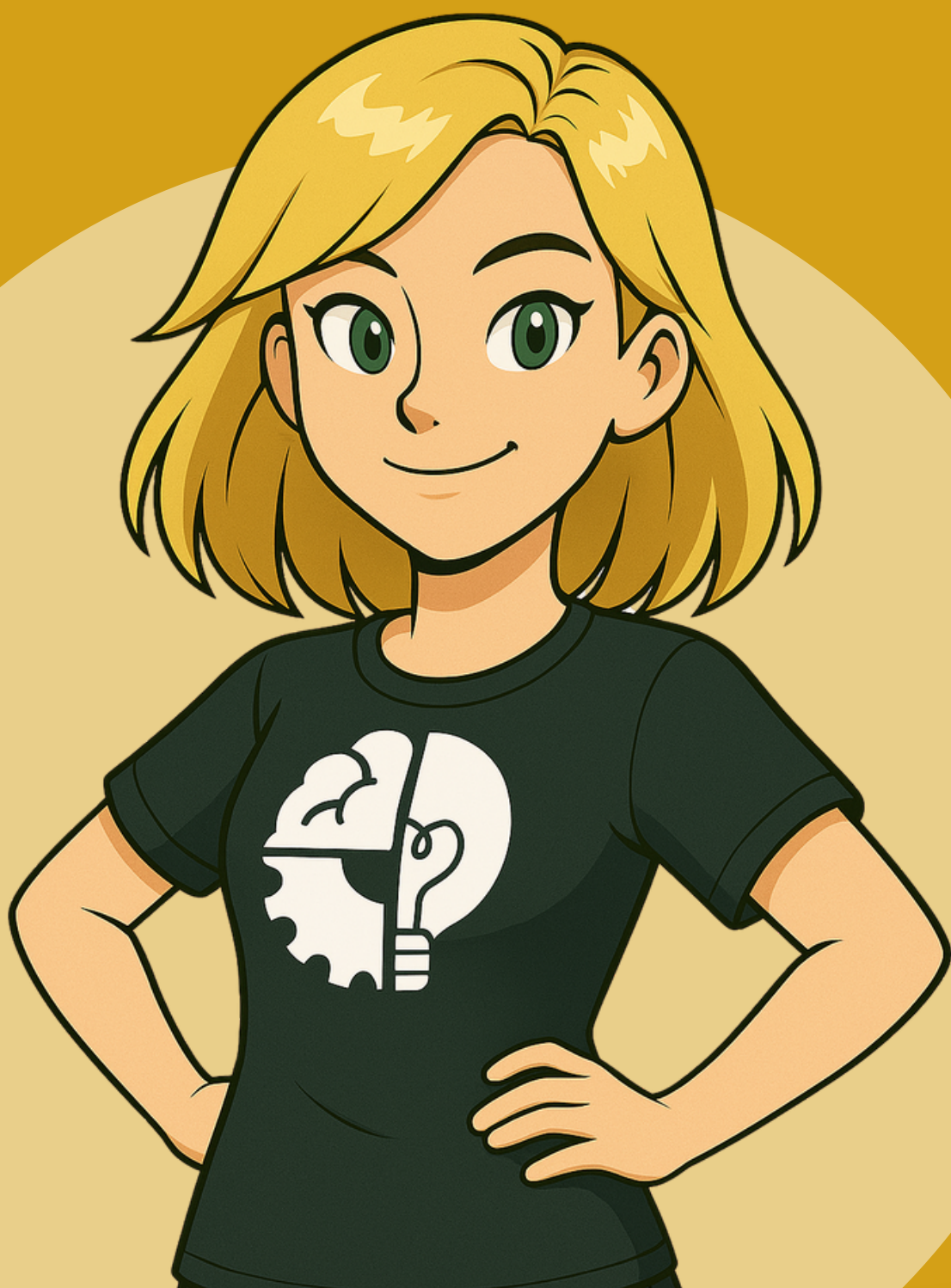


Elodie TURLIER

Angular, IA, agilité & storytelling



ABONNE TOI POUR PLUS DE CONTENU



Elodie TURLIER

Angular, IA, agilité & storytelling : construire des apps prêtes à survivre à l'apocalypse digitale



Enregistre ou partage si ça t'a été utile