

Exercice 01 — Bases de la librairie Angular

Objectif

Mettre en place une librairie `shared-ui` prête à accueillir des composants réutilisables.

Étapes

1. Générer la librairie

```
ng generate library shared-ui --standalone
```

- Vérifiez que `projects/shared-ui/src/public-api.ts` exporte un module racine.
- Activez `changeDetection: ChangeDetectionStrategy.OnPush` dans le composant par défaut.

2. Configurer les paths TypeScript

```
// tsconfig.base.json
{
  "compilerOptions": {
    "paths": {
      "@shared/ui": ["projects/shared-ui/src/public-api"],
      "@shared/ui/*": ["projects/shared-ui/src/lib/*"]
    }
  }
}
```

3. Créer un module utilitaire

- Ajouter `SharedUiModule` qui exporte les composants du dossier `lib`.
- Prévoir un `SharedUiConfig` (`InjectionToken`) pour gérer futur theming.

4. Initialiser Storybook (si non présent)

```
npx storybook@latest init --type angular
```

5. Build & lint

```
ng build shared-ui
npm run lint shared-ui
```

Résultats attendus

- La librairie compile (`dist/shared-ui`).
- Les composants sont accessibles via `import { ... } from '@shared/ui';`.
- Storybook peut charger les composants de la lib.

Critères de réussite

- Aucun `ChangeDetectionStrategy.Default` laissé par défaut.
- Un README minimal (`projects/shared-ui/README.md`) décrit les conventions.
- Les composants utilisent `standalone: true` + `imports` explicites.

Passez à l'exercice 02 pour factoriser les formulaires.