

Exercice 04 — Tests unitaires & accessibilité

Objectif

Garantir la robustesse et l'accessibilité des composants partagés via des tests ciblés.

Étapes

1. Mettre en place un setup de tests dédié

```
// projects/shared-ui/src/testing/testbed.setup.ts
export function setupSharedUiTesting() {
  TestBed.configureTestingModule({
    providers: [provideNoopAnimations()],
  });
}
```

2. Tester `shared-form-field`

- Vérifier l'ajout automatique de `aria-describedby`.
- S'assurer que l'erreur ne s'affiche qu'en cas de `touched && invalid`.

```
it('link error message to input via aria-describedby', () => {
  const errorId = fieldEl.getAttribute('aria-describedby');
  expect(errorId).toContain('shared-form-field-error');
});
```

3. Tester `shared-select (CVA)`

- `writeValue` applique la valeur.
- `setDisabledState` désactive l'input + applique `.is-disabled`.
- `onChange` est déclenché lors d'une sélection clavier.

4. Storybook + axe-core

- Ajouter Storybook addon ally (`npm install @storybook/addon-ally`).
- Corriger erreurs (contraste, label, focus) jusqu'à avoir 0 violation.

5. Check accessibilité

- Compléter `checklists/checklist-accessibilite.md`.

Résultats attendus

- Minimum un test par composant critique (button, input, form-field, select).
- Couverture > 80% sur `projects/shared-ui/src/lib/`.
- Rapport axe-core "no violations" sur stories principales.

Bonus

- Intégrer tests screenshot (Chromatic ou Playwright) pour valider le rendu.
- Injecter `logger` pour tracer `ControlValueAccessor` en dev.

Référez-vous à `exercices/solutions/04-testing.md` pour le corrigé.