

Corrigé

P1C5 - Vérifiez le front avec React Testing Library



Commençons par préparer le mock :

```
TypeScript
const onSubmit = vi.fn();
```

Nous affichons ensuite le composant :

```
TypeScript
render(
  <TaskForm onSubmit={onSubmit} />
);
```

Le premier test vérifie le rendu initial :

```
TypeScript
expect(
  screen.getByRole("textbox", {
    name: /titre/i
  })
).toBeInTheDocument();

expect(
  screen.getByRole("button", {
```

```
    name: /ajouter/i  
  })  
).toBeVisible();
```

Nous simulons ensuite les interactions :

```
TypeScript  
const user = userEvent.setup();  
  
await user.type(  
  screen.getByRole("textbox", {  
    name: /titre/i  
  }),  
  "Nouvelle tâche"  
);  
  
await user.click(  
  screen.getByRole("button", {  
    name: /ajouter/i  
  })  
);
```

Enfin, nous vérifions l'appel du callback :

```
TypeScript  
expect(onSubmit)  
  .toHaveBeenCalledWith({
```

```
title: "Nouvelle tâche"  
});
```

Pour maintenir un fichier lisible, ces tests peuvent être regroupés dans des blocs `describe()` distincts dédiés au rendu initial et aux interactions utilisateur.

Grâce à cette couverture, votre tech lead sait désormais que le formulaire fonctionne correctement du point de vue de l'utilisateur et qu'il transmet les informations attendues au reste de l'application.