

# Corrigé

P1C3 - Écrivez des tests unitaires



Voici une solution possible :

```
TypeScript
import { describe, it, expect } from "vitest";
import { hashPassword } from "../hash";

describe("hashPassword", () => {

  it("should return a hashed string when given a valid password",
  async () => {

    const password = "MonMotDePasse123";
    const result = await hashPassword(password);

    expect(result).not.toBe(password);
    expect(typeof result).toBe("string");

  });
});
```

```
it("should return a hashed string when given an empty password",
  async () => {

    const password = "";

    const result = await hashPassword(password);

    expect(typeof result).toBe("string");

  });

it("should hash a very long password", async () => {

  const password = "a".repeat(1000);

  const result = await hashPassword(password);

  expect(typeof result).toBe("string");

});

});
```

Lancez ensuite :

```
Shell
npm run test
```

Les trois tests doivent passer au vert.

Pour provoquer un échec volontaire, remplacez :

```
TypeScript  
expect(result).not.toBe(password);
```

par :

```
TypeScript  
expect(result).toBe(password);
```

Relancez Vitest.

Le terminal affiche alors un message indiquant :

- le test concerné ;
- la ligne en échec ;
- la valeur attendue ;
- la valeur reçue.

Une fois l'analyse terminée, remettez l'assertion correcte et vérifiez que l'ensemble de la suite de tests redevient vert.

Vous venez de réaliser votre première mission de couverture de test dans l'application.

Votre tech lead dispose désormais d'un filet de sécurité qui permettra de faire évoluer le service d'authentification avec davantage de confiance.