Övning: Mocka API-anrop i React-test med Mock Service Worker (MSW)

Vi testar vår **DataList-komponent**, som hämtar användardata från `jsonplaceholder.typicode.com`. Vi använder **Mock Service Worker (MSW)** för att mocka API-anrop.

Installera MSW

Kör följande kommando för att installera Mock Service Worker:

```
npm install --save-dev msw
```

Skapa mock-server i `src/mocks/server.ts`

Skapa en fil `src/mocks/server.ts` och lägg till följande kod:

```
import { setupServer } from "msw/node";
import { http, HttpResponse } from "msw";

export const server = setupServer(
  http.get("https://jsonplaceholder.typicode.com/users", async () => {
    //Data som returneras här kan användas i alla test
    //Alternativt kan testen overrida med annan data
    return HttpResponse.json([
      { id: 1, name: "Test User" }
    ], { status: 200 });
  })
);
```

Förklaring `server.ts`

- `setupServer` skapar en **mock-server** som fångar upp nätverksanrop.
- `http.get(...)` fångar GET-anrop till API:et och returnerar en mockad respons.
- Vi returnerar default en **testanvändare**, vilket gör att vi slipper konfigurera extra mock-data i testfilen.

Uppdatera `setupTests.ts`

Lägg till mer kod i `src/setupTests.ts`:

```
import { server } from "./mocks/server";
import { beforeAll, afterEach, afterAll } from "vitest";

// Starta MSW innan testerna körs
beforeAll(() => server.listen());

// Återställ eventuella anpassade handlers efter varje test
afterEach(() => server.resetHandlers());

// Stäng servern efter att alla tester har körts
afterAll(() => server.close());
```

Skapa test för `DataList` i `src/tests/DataList.test.tsx`

Skapa filen `src/tests/DataList.test.tsx` och lägg till följande kod:

```
import { render, screen, waitFor } from "@testing-library/react";
import DataList from "../components/DataList";

// Testa att komponenten hämtar och visar användare
test("visa users från jsonplaceholder", async () => {

  render(<DataList />);

  // Vänta på att datan ska laddas och synas i DOM:en
  await waitFor(() => {
    expect(
      screen.getByText("Test User")
    ).toBeInTheDocument();
  });
});
```

Vad gör denna testfil?

1. **Renderar `DataList`** → Vi testar komponenten utan att göra riktiga API-anrop.
2. **Väntar på att datan ska laddas** → Eftersom API-anrop är asynkrona, använder vi `waitFor()`.
3. **Kontrollerar att användaren syns i DOM:en** → `expect(screen.getByText("Test User")).toBeInTheDocument()` ser till att den mockade användaren visas.