FileLoaderComponent

## Framework

Angular

# Goal

Test a component that loads data from a service using TDD. Replace the actual service with a stub to isolate the unit and ensure deterministic behaviour.

# Instructions

1. Create a FileService with a method fetchFile(id: string) that returns an Observable.
2. In your test file, create a stub for FileService using jest.fn().mockResolvedValue(...).
3. Provide the stubbed service using TestBed.configureTestingModule and useValue.
4. Write a test that renders the component with a given fileId, triggers change detection, and waits for the stubbed data to render.
5. Write another test that returns an error or empty result and assert that fallback content is displayed.
6. Use fixture.detectChanges() and fixture.whenStable() to ensure async code completes before assertions.

# Tech & Tools

* Angular TestBed and ComponentFixture
* of() and throwError() from rxjs to simulate responses
* Jest mocks and spies

# What to Test

* Component displays “Loading...” initially
* Stubbed file content is rendered after data fetch
* Fallback UI appears on failure or empty response
* Real HTTP requests are not triggered