Angular Testing with Jest

Khalil Themri

Chapter 1

Introduction to Jest Testing

Running Jest Files in Visual Studio Code

Before diving into the examples, it's important to know how to run Jest tests in your development environment. If you're using Visual Studio Code (VSCode), there are a couple of plugins that can enhance your experience:

1. **Jest Plugin by Facebook:**

   - Install the *Jest* plugin by Facebook from the VSCode extensions marketplace.
   - This plugin automatically detects Jest tests in your project and runs them in the background. You can see the test results directly in the editor.
   - It provides helpful features like inline test results and coverage visualization.

2. **Jest Runner:**

   - Install the *Jest Runner* plugin from the VSCode extensions marketplace.
   - Jest Runner allows you to run or debug a single test or a whole test file with just a click. This is particularly useful when you're working on a specific part of your test suite.
   - To use it, right-click on a test file or test function and select "Run Jest" or "Debug Jest".

With these tools, running and debugging your Jest tests becomes much more efficient, allowing you to focus on writing and improving your tests.

Basic Jest Examples

Let's start with some simple Jest examples to understand the fundamentals of testing.

Listing 1.1: A super basic test example.

```
describe('super basic test', () => {
  it('true is true', () => {
    expect(true).toEqual(true);
  });
});
```

The above example is a basic test that simply checks if 'true' is equal to 'true'. This is useful for ensuring that your testing environment is set up correctly.

Listing 1.2: Testing the Cat class getters and setters.

```
import { Cat } from './cat';

describe('Test Cat getters and setters.', () => {
  it('The cat name should be Gracie', () => {
    const cat = new Cat();
    cat.name = 'Gracie';
    expect(cat.name).toEqual('Gracie');
  });
});
```

In this example, we test a simple getter and setter in a 'Cat' class to verify that the 'name' property is set and retrieved correctly.

In this example, tests are grouped by chapters, which allows for a clear organization of tests as you progress through different sections or chapters of your project or tutorial.

Chapter 2

Testing the Contact Class

Now that we've covered the basics, let's move on to a more practical example where we test a ‘ContactClass‘. This will involve testing various methods and properties within the class.

Listing 2.1: Test cases for the `ContactClass`.

```
import ContactClass from './contact';

describe('Contact class tests', () => {
  let contact: ContactClass | null; // Declares the contact variable as
      a ContactClass type

  beforeEach(() => {
    // Executes beforeEach function before each test case
    contact = new ContactClass();
  });

  it('should have a valid constructor', () => {
    // Tests that the contact is not null
    expect(contact).not.toBeNull();
  });

  it('should set name correctly through constructor', () => {
    contact = new ContactClass('barthauer');
    expect(contact.name).toEqual('barthauer');
  });

  it('should get and set id correctly', () => {
    contact!.id = 1;
    expect(contact!.id).toEqual(1);
  });

  it('should get and set name correctly', () => {
    contact!.name = 'barthauer';
    expect(contact!.name).toEqual('barthauer');
  });

  it('should get and set email correctly', () => {
    (contact as any)!.email = 'barthauer@barthauer.de'; // Casting to
         any to test dynamic property
    expect((contact as any)!.email).toEqual('barthauer@barthauer.de');
  });

  afterEach(() => {
    // Executes afterEach function after each test case
    contact = null;
  });
});
```

By organizing your tests using `describe`, `it`, `beforeEach`, and `afterEach`, you create a clear, maintainable, and effective testing structure. Each test is isolated, ensuring that one test's setup or outcome doesn't affect another. The use of `.spec` suffixes and component names in filenames further enhances the readability and manageability of your test suite.