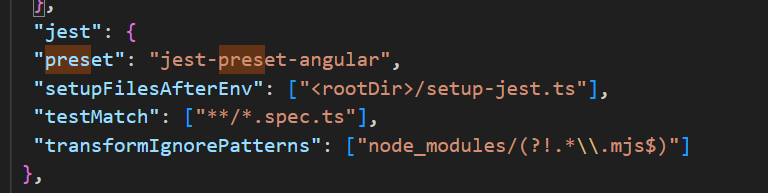
Jest Framework has **Snapshot testing** support.

**Faster execution** (runs outside a browser).

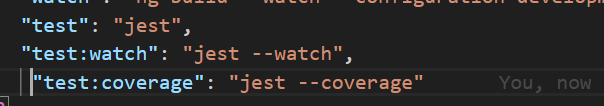
First, install Jest, jest-preset-angular, and other required dependencies:

npm install --save-dev jest jest-preset-angular @testing-library/angular @types/jest ts-jest

Modify package.json to use Jest instead of Karma:



Modify the test script to use Jest instead of Karma:



Now, you can run your tests with:

npm test

For **watch mode** (auto-runs tests on file changes):

npm run test:watch

For **code coverage report**:

npm run test:coverage

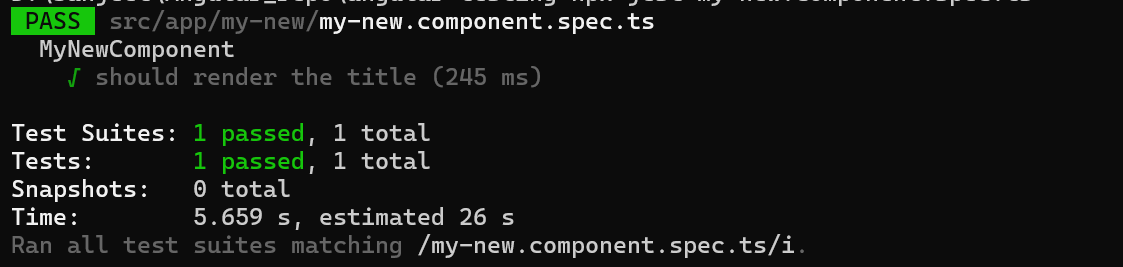
To test only one component

Run **only one test file** using:

npx jest my-new.component.spec.ts

or using npm

npm test -- my-new.component.spec.ts



**Jasmine test cases need some modifications to work with Jest** because:

* **Jasmine uses spyOn(object, 'method')**, while **Jest uses jest.spyOn(object, 'method')**.
* **Jasmine has beforeEach, afterEach, and matchers like toHaveBeenCalled()**, which are also in Jest, but some assertions work differently.
* **Karma auto-detects changes**, while **Jest runs tests in isolation**.

**Converting Jasmine to Jest**

Here’s what you need to change in your Angular test files:

spyOn(service, 'getUser').and.returnValue(of(mockUser));

change to Jest:

jest.spyOn(service, 'getUser').mockReturnValue(of(mockUser));

**Replace createSpyObj**

**Jasmine:**

const httpClientSpy = jasmine.createSpyObj('HttpClient', ['get']);

change to Jest:

const httpClientSpy = jest.spyOn(global, 'HttpClient');

**Convert done Callbacks for Async Tests**

**Jasmine:**

it('should fetch data', (done) => {

service.getData().subscribe((data) => {

expect(data).toEqual(mockData);

done();

}); });

Change to Jest with async/await:

it('should fetch data', async () => {

await expect(service.getData().toPromise()).resolves.toEqual(mockData);

});

Most Jasmine matchers work in Jest (toBeTruthy(), toHaveBeenCalled(), etc.), but some need tweaks:

