

Testing Node.js Applications

Node.js Accelerator – Jan 23

Agenda

1. Importance of test

2. Unit vs E2E

3. E2E with supertest

4. Writing Tests and Assertions

5. Testing an Express Application



Node.js Accelerator

Jan 2023

Some resume of reasons why testing is important:

- Reduce bug fixing cost
- Avoid non necessary production issues
- Continuous integration
- Regression testing
- Increase of quality of the software

Both unit and e2e are important when working with an application. Meanwhile unit test can isolate the work you are doing, e2e testing allows you to test a full functionality of what you are building.

Working with microservices allows the e2e testing to be a bit easier to maintain than in a monolithic way that could go for minutes of e2e running in the CI/CD

When working with unit test, ideally you should avoid going deeper more than one level of what you are testing and mocking all the rest. Unit testing tends to make your code design more consistent and testable

More on this [link](#)



When working with e2e, it is a better idea to isolate the application and server in express.

Exporting the express application allows you to use supertest to run e2e integrations

- We can either try to connect directly to external API's when doing e2e and/or mocking them for your local tests

```
test( testName: "It should response the GET method", fn: async () => {  
  const response = await request(app).get( url: "/hello");  
  expect(response.body.msg).toBe( expected: 'Hello World')  
  expect(response.statusCode).toBe( expected: 200);  
});
```

Using JEST library we can define a set of functions with describe and test and then define what we need to run

```
11 describe( name: 'UserService', fn: () => {  
12   test( testName: "Should be defined", fn: async () => {  
13     expect(userService).toBeDefined();  
14   })  
15  
16   test( testName: 'Expect putUserById to receive userId', fn: async ()=>{  
17     const response = await userService.putUserById(userId);  
18     expect(response.userId).toBe(userId)  
19   })  
20 });
```

Let's see our office hours express code and do some tests there!

1. We strongly recommend you to explore below module through Udemy this week and learn about Node.js essentials

Module	Topic	Description	Udemy Link	
Self-paced learning on Udemy	Deployment	Learn about deploying Node.js applications on heroku. Setup git, prepare app for deployment and finally deploy on heroku	Click here	1 hrs 2 min

