

Testing Node.js Applications part 2

Node.js Accelerator – Jan 23

Agenda

1. Test with Authentication

2. Code Coverage

3. Express: code coverage with jest

4. Coding tests with NestJS

When using Auth0, for example, we sometimes need to automate token generation for a testing user.

For allowing this feature, we need to do some changes in the Auth0 dashboard with API authorization settings and a patch to the application id:

https://auth0.com/docs/api/management/v2#!/Clients/patch_clients_by_id

Lets see it on the dashboard and our code

API Authorization Settings

Default Audience

API Audience to use by default for [API Authorization flows](#) . Note: This setting is equivalent to appending the audience to every authorization request made to the tenant for every application. This will cause new behavior that might result in breaking changes for some of your applications. If you require assistance, [contact support](#).

Default Directory

Name of the connection to be used for Password Grant exchanges. The `default_directory` value should be the exact name of an existing connections of one of the following strategies: ad, auth0, email, sms, waad, adfs

Patching sample:

```
{
  "grant_types": [
    "password",
    "http://auth0.com/oauth/grant-type/password-realm",
    "client_credentials",
    "authorization_code",
    "implicit",
    "refresh_token"
  ]
}
```

When working with automated tests, usually the tools such as Jest allows the code coverage by default

It is important to define the files you want to cover and the minimum percentage of the coverage you need for your project. When combined with CI/CD you can block a branch to be merged if it didn't get the minimum requirements.

When working with a project that doesn't have any tests, you can add the tests and set the criteria for new code to be tested and to setup a spare time with the team to try to increase the test coverage from the legacy code in sprints.

Let's see the code coverage on our current office hour project

Let's see now how to code tests in NestJS environment

A suggestion now is to try exploring some way to use the tests in your CI/CD pipeline (we will have an office hour for it) and to use automated tools for code quality such as code climate and/or sonar cloud

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

Module	Topic	Description	Udemy Link

