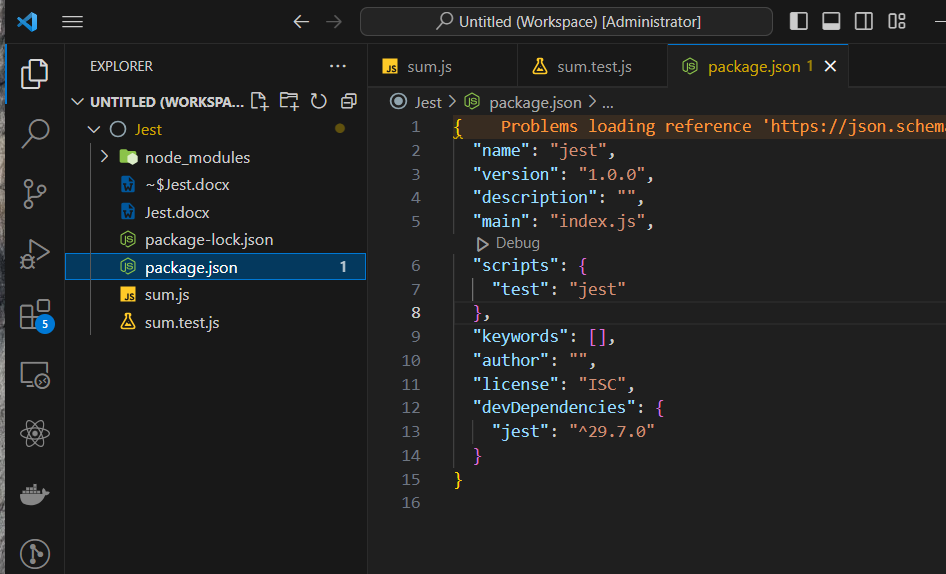
First initialize an npm project

* npm init -y : -y is a shortcut to say yes to all the prompts

**To install Jest**

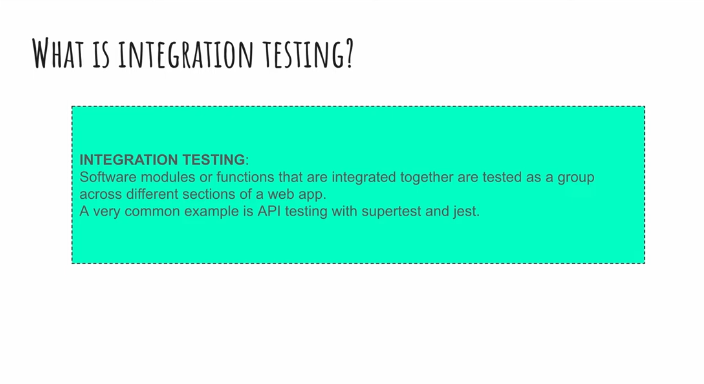
* npm install –save-dev jest

Modify the package.json scripts to run tests with Jest

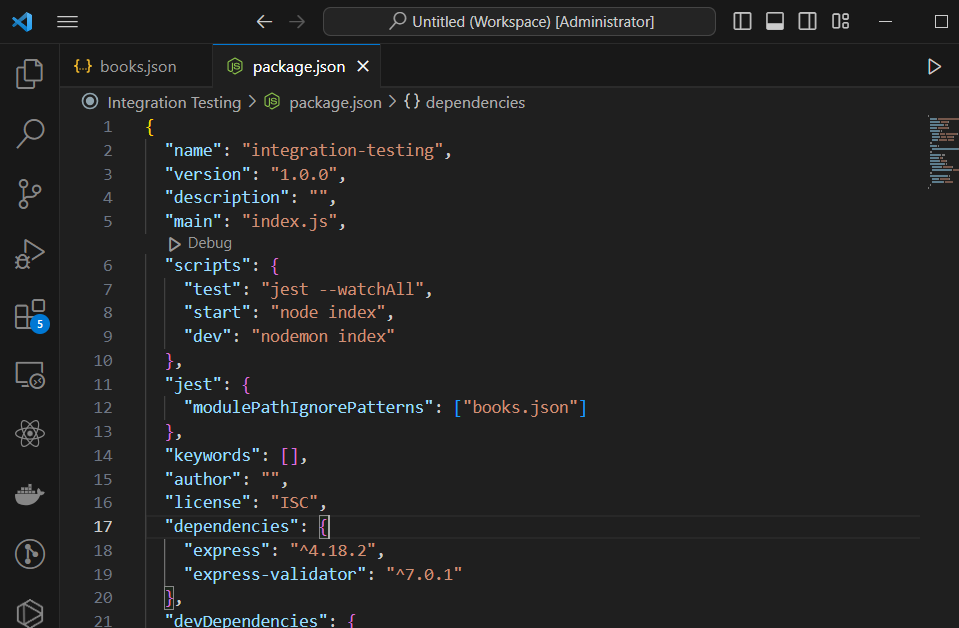


* Unit testing – testing small blocks of code like basic functions and basic classes. Helps us identify errors early in the development phase.
* To run a test, type the command **npm test**
* Matcher includes functions like toBe(), toEqual(), toFalsy(), toTrue();
* toBe() is basically used for primitive values like numbers, strings and Booleans.
* If we’re dealing with objects and arrays we say toEqual().
* Mocks are fake implementations of real functions and spies are tools used to track the behavior of those functions.

**Integration Testing**

****

* npm i express express-validator
* npm i -D jest supertest
* npm i -g nodemon

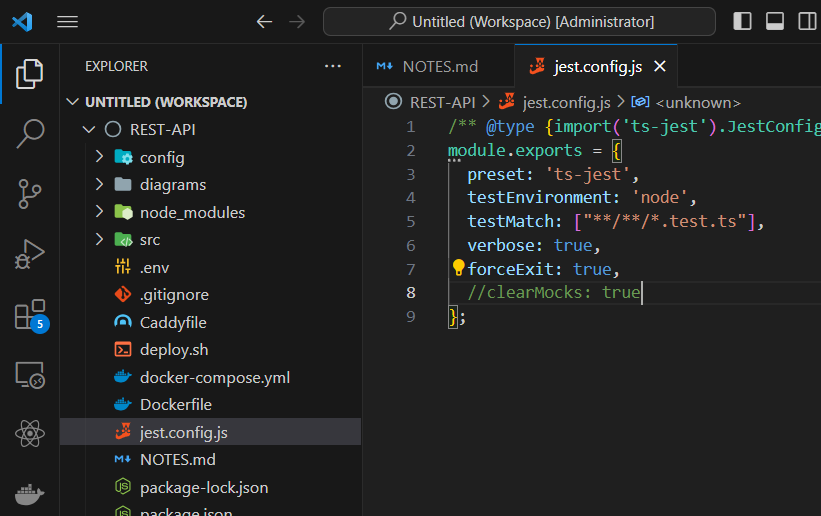


* npm run dev – starts the server with nodemon

**NOTE:** Only test modules you created on your own

PART THREE

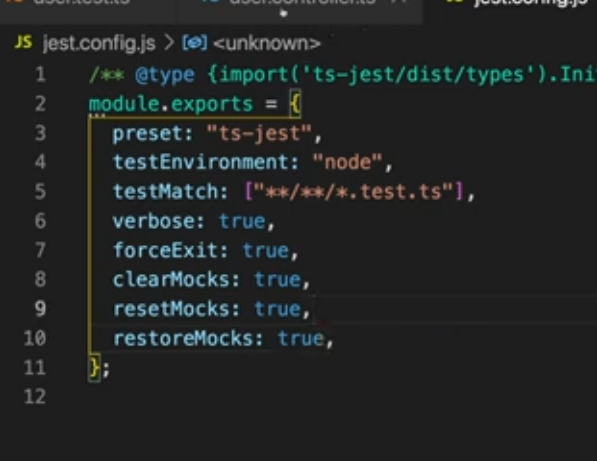
* yarn ts-jest-config:init to initialize a jest config file



Some pre-configurations to use.

NOTE:

* expect.any is used when you’re receiving dynamic data.
* When testing it’s always good to sit back and list down the things you’re going to test.



This clear, reset and restore mocks configuration allows the mocks to be reset before proceeding to the next to avoid collisions.