

Introduction to Cypress

Given a workshop this morning to try and use Cypress from the docs (no lecture)

Uses describe blocks like in Jest (same syntax as Jest – human readable statement then a function to contain the test(s))

Then uses “it” instead of “Test” (same syntax again)

Use `cy.visit(“URL”)` to make it visit a page

Supertest

Used for testing HTTP

Can be used on its own or combined with another testing suite – we will be using Jest with it today. This is because supertest isn’t great at giving you feedback on the tests whereas Jest logs it to the console.

Could be used as an alternative to automate the tests we were doing with Postman to test our APIs were working (sending requests and checking the response). (Postman does have automated capabilities and isn’t being replaced by this).

Read the docs! <https://www.npmjs.com/package/supertest>

Basic things it can check:

<code>‘Content-type’, /json/</code>	Very important - ensure we’re getting a json back
<code>‘Content-Length’, ‘15’</code>	Less important – just checking the length of what you get back (doesn’t have to be absolute, could just be > 0?)
<code>‘200’</code>	Checks HTTP status code – 200 is success with no errors

`.expect` is a method you are calling on the object that we are being given back after we’ve run:

`request(app)`

`.get(‘/user’)`

<code>.expect(‘Content-Type’, /json/)</code>	Reg ex (regular expression) for pattern match in strings This is just checking if the thing includes/contains json
--	---

Expect can take different arguments depending on what you’re testing (see API section at bottom of docs)

Use the mocha examples which work very similarly to Jest to see how to incorporate Supertest into Jest.

Don’t worry about “done” keyword anywhere – this is from an older version of JS that is now obsolete as we now tend to use `async/await` version of the code that is in the docs.