

Using Mocks



Gabriel McNeilly

QA Automation Lead

@gmcneilly software-tester.io

Mocks



Allow us to test the links between code



Capture function calls



Allow for configuration of return values



Help avoid slow and unreliable tasks in tests



Allow us to just test what we need to test





Module Overview

Creating simple mock functions

Automock

Creating manual mocks



Creating Simple Mock Functions



Automock



Automock



All imported modules will be automatically mocked



Removes the need to add explicit mocks for dependencies



You must “unmock” any functions if you want to use their actual implementation



Can help to ensure your tests are only testing a single “unit”



Can be difficult to introduce into existing test suites



Creating Manual Mocks



Key Takeaways

Mocks

- Used to stub out functionality with static data to ensure tests are fast and not flaky

Mock functions

- Used to mock individual functions within your tests

Automock

- Toggled on via your Jest config file
- By default, automatically mocks all imported modules except for built-in Node.js modules

Manual mocks

- Used to mock module dependencies
- Kept in `_mocks_` directories

