

# Testing Advanced Functionality with Jest

---



**Daniel Stern**

Github – [danielstern](#) | [@danieljackstern](#)



# Leveraging Advanced Mock Functionality

Mocks are not the same as stubs

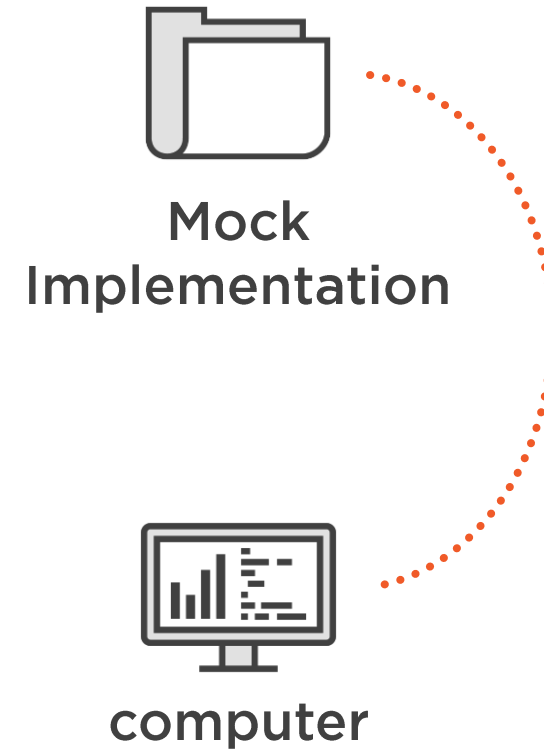
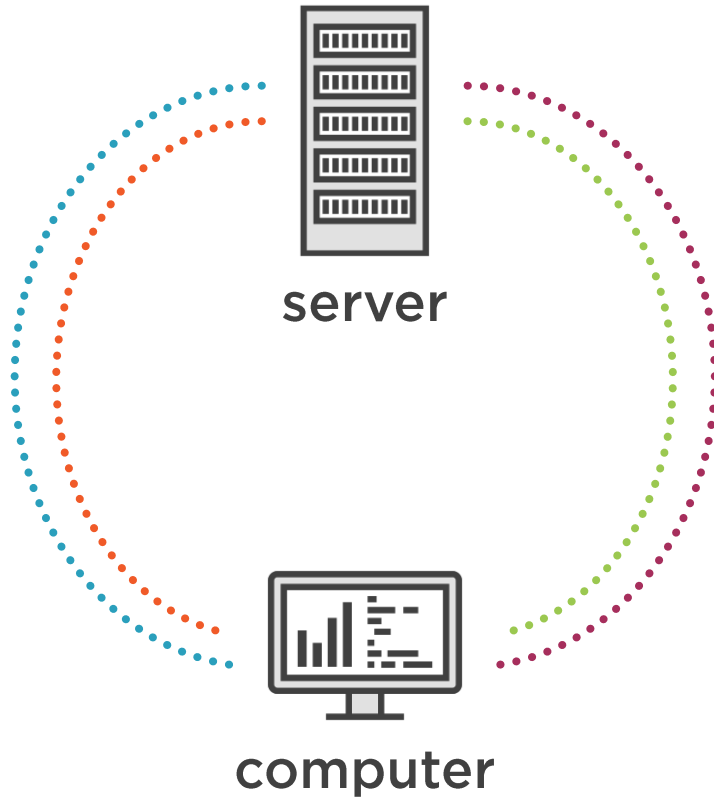
A stub is a mostly-empty object with no functionality

A mock is a similar stub but has lots of test-related functionality, e.g., counting function calls

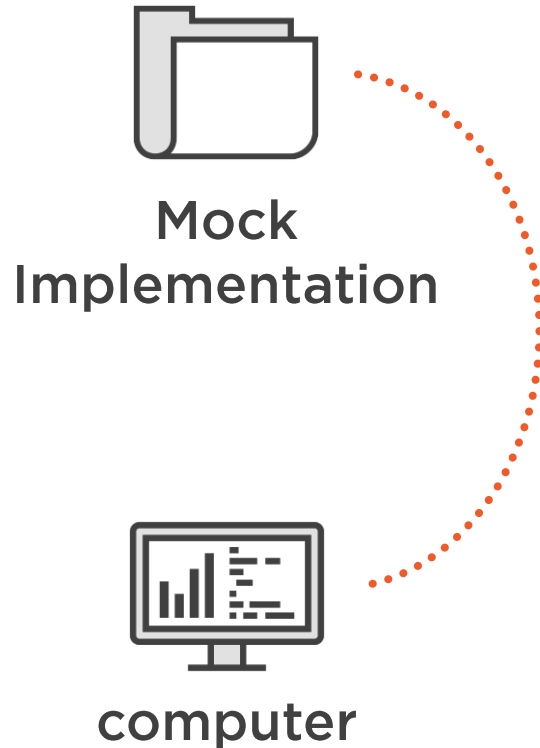
Use a mock to verify a function has been called with the correct arguments



# Mock Implementations



# Mock Implementations



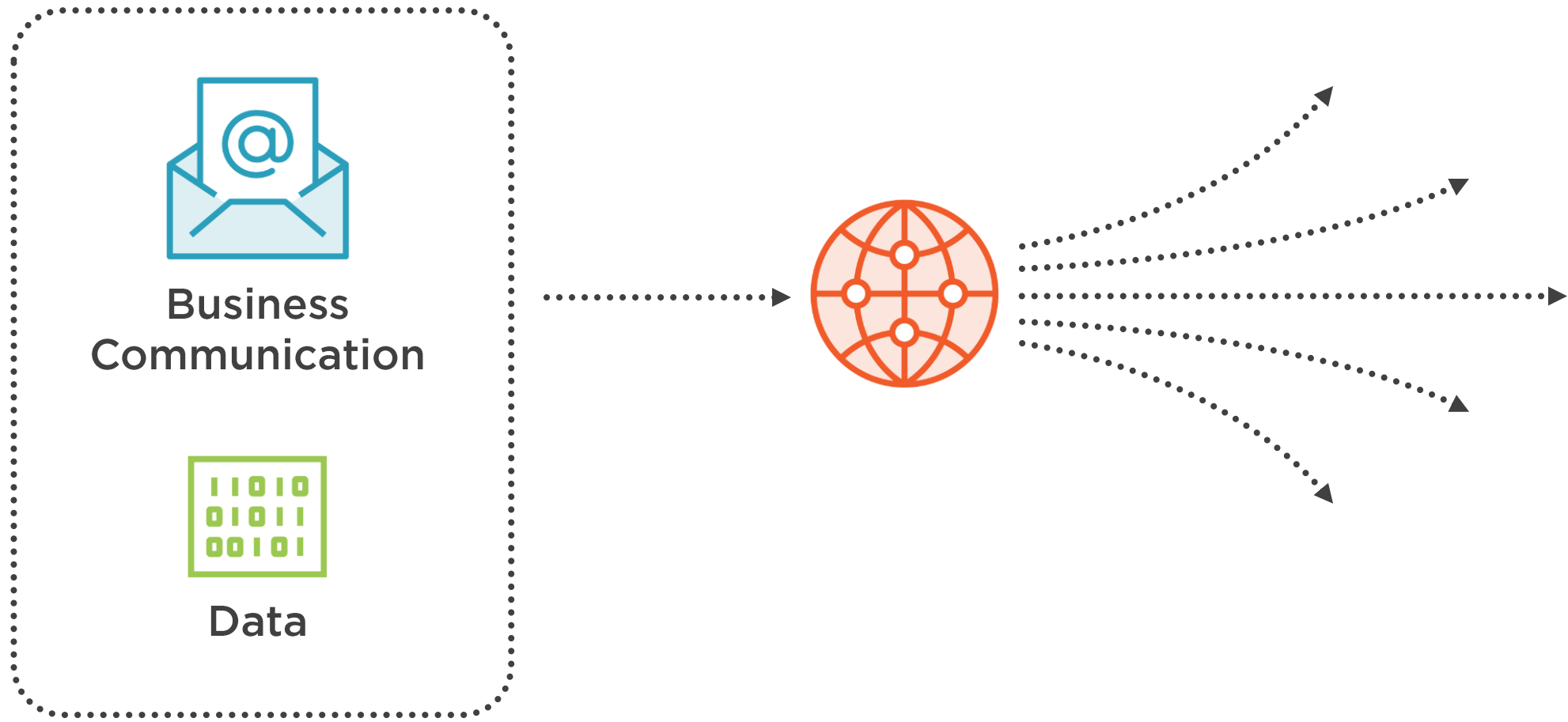
Create custom code that replaces the actual code for a mocked function

Useful for passing a controlled value to a component in order to test the output

We will create a mock implementation that returns a controlled value, allowing a consistent test



# Testing HTTP Requests - Sending Requests



# Testing HTTP Requests Sending Requests

dealing with HTTP requests

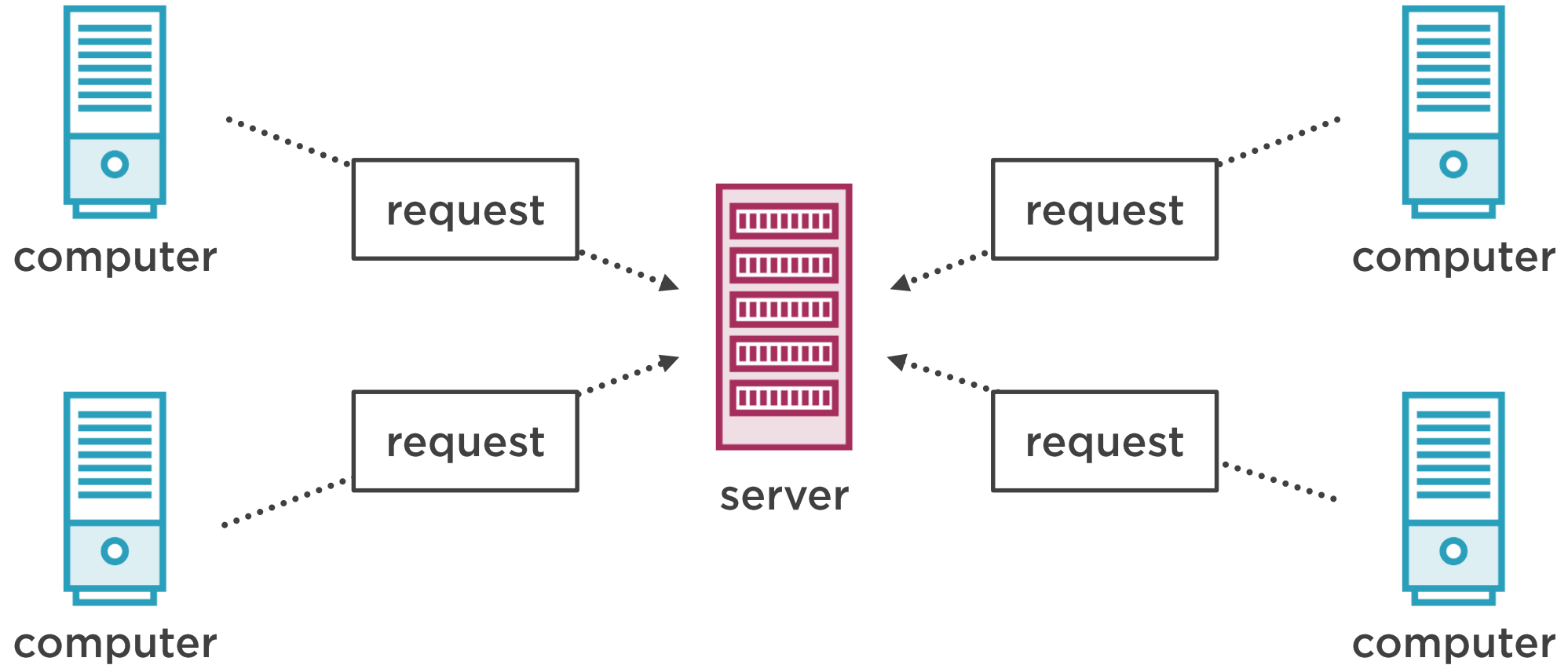
using dev tools or proxies is error- prone

difficult to verify if the correct code  
was sent to the server

Jest “catches” requests with a mock  
allowing test/verify code



# Testing HTTP Requests - Handling Responses



# Testing HTTP Requests Handling Responses

Websites must respond gracefully to many kinds of HTTP responses

200 OK, 404 Not Found, and 500 (Internal Server Error) must all be accounted for

Hard to simulate responses in browser

Easy to simulate in tests

Often only need to verify that a request has been made to correct URL





# Measuring Test Coverage



**Test Coverage is a test for your tests**

**Hard to verify application works without tests; hard to verify tests are doing anything without coverage**

**Istanbul, a popular coverage tool for JavaScript, is supported by default for Jest**

**By installing the appropriate packages and updating our `package.json`, we will enable Test Coverage**

