

An In-Depth Look at the Jest API



Daniel Stern

Github – danielstern | @danieljackstern





Most Jest API methods relate to automatic mocking

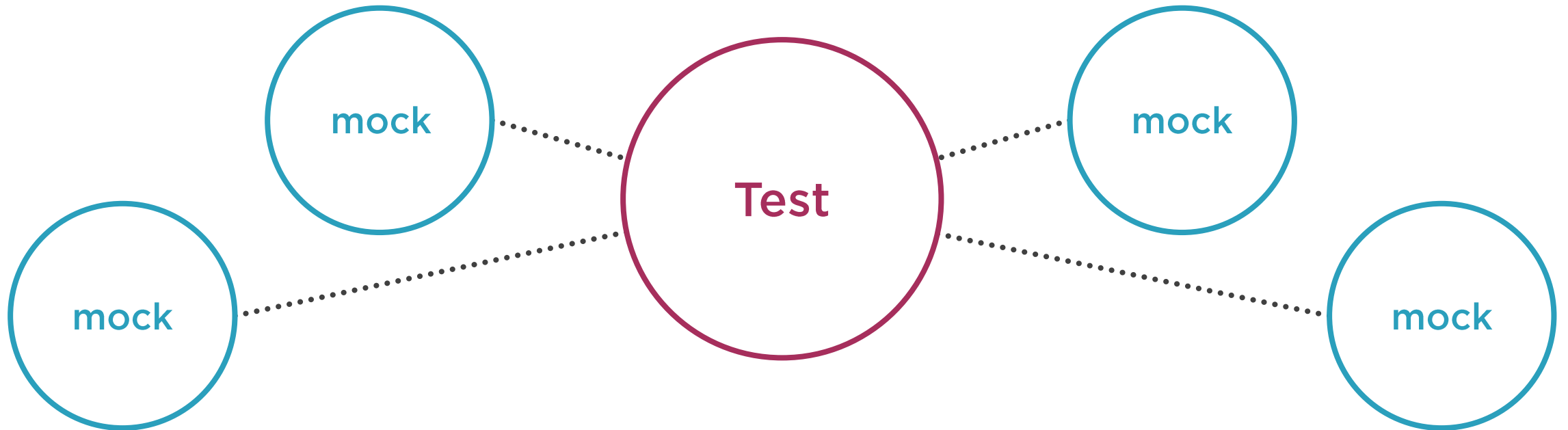
``dontMock`` is a common and important method

Most test logic still uses Jasmine interface

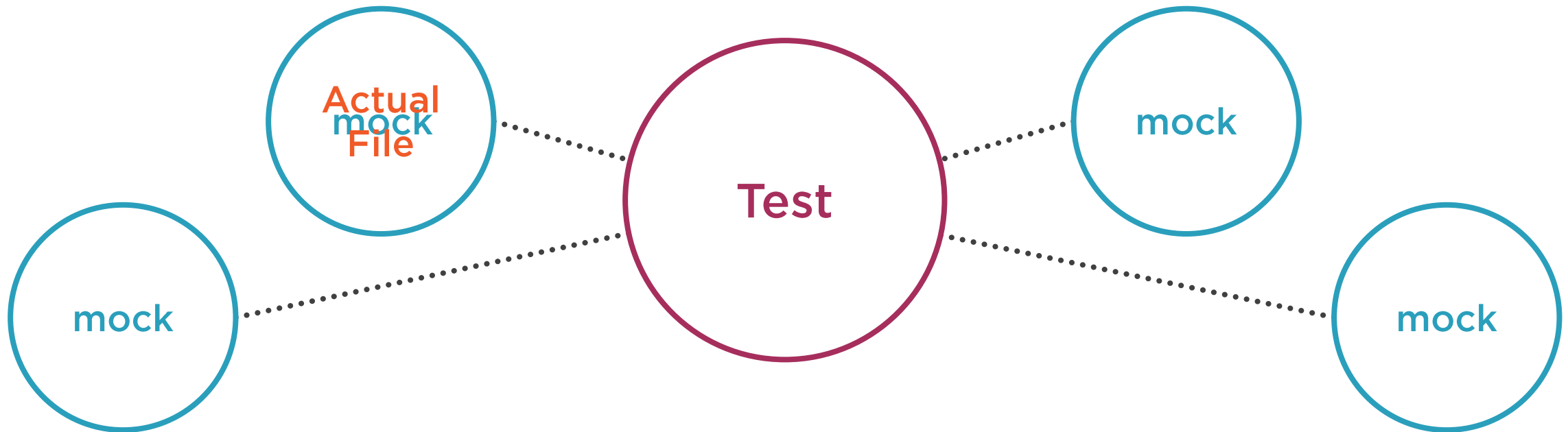
Generally the Jest API allows users to create, modify and analyze mocks



Disabling Mocking for A Single Module



Disabling Mocking for A Single Module



```
[jest.dontMock("myModule")]
```



Libraries (i.e./ jQuery, Underscore) are usually not mocked

Generally, the libraries for which mocking is disabled are defined in 'package.json'

Try to avoid using 'dontMock' with libraries



Toggling Automatic Mocking

`jest.autoMockOff()`

Disables automatic mocking

`jest.autoMockOn()`

Enables automatic mocking
(default, recommended)



Turning automatic mocking off is the same as calling 'dontMock' on every dependency



Useful for tests with large number
of unmocked modules

(“implementation details”)

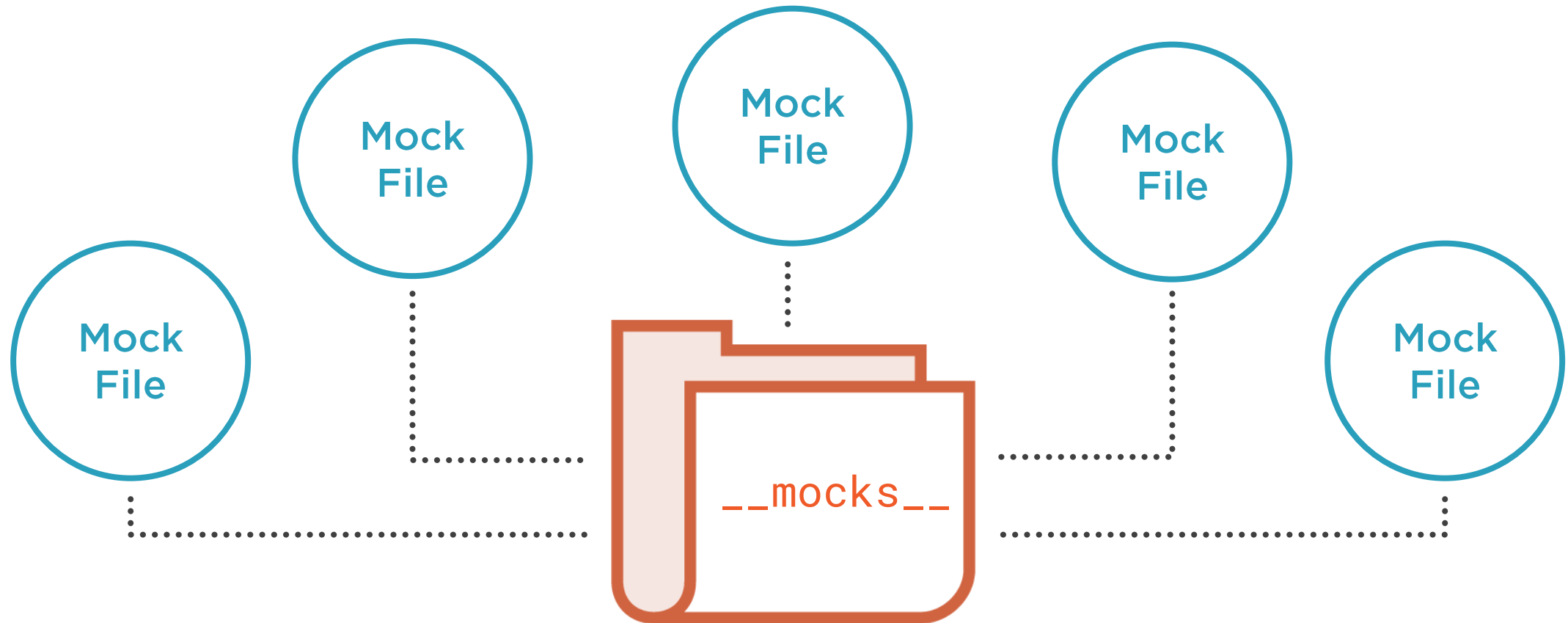
``jest.mock()`` is the reverse of
``jest.dontMock()``

Source:

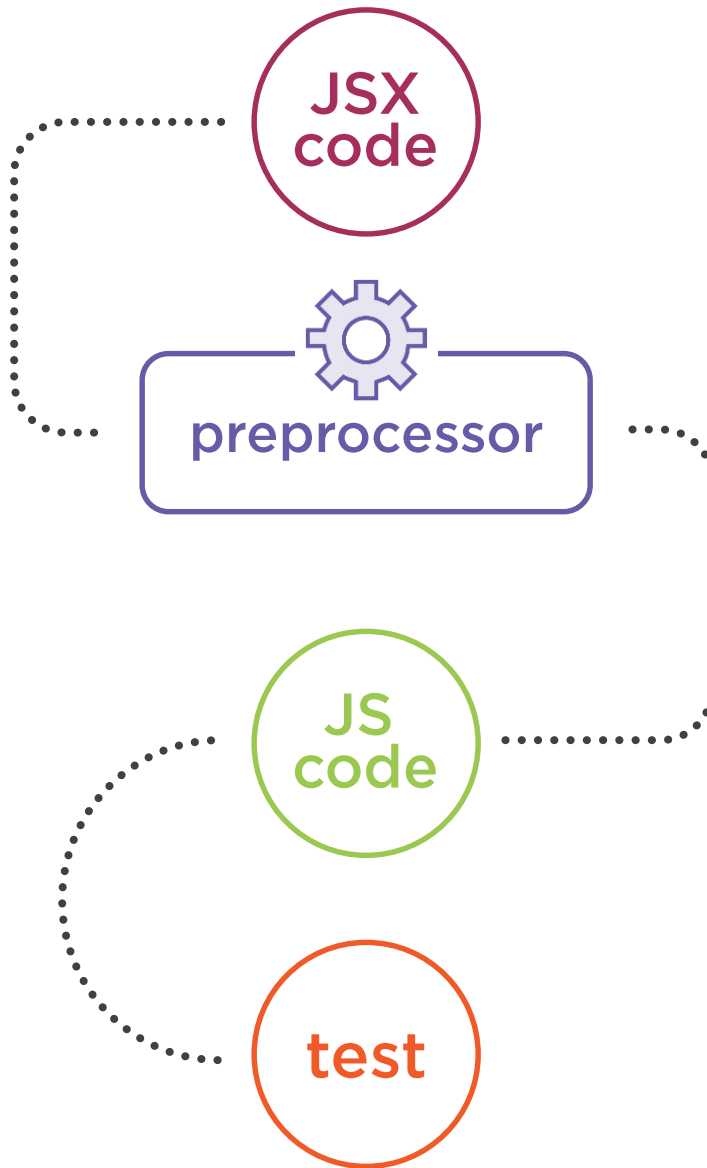
<https://facebook.github.io/jest/docs/api.html#jest-genmockfunction>



Specifying Manual Mocks



Defining a Preprocessor



JSX is not natively supported by Node 4.0

Preprocessors transform JSX into JavaScript

Babel (an ES6/ES7/ESNext preprocessor) also processes JSX



“[Jest is] built on top of the Jasmine test framework, using familiar `expect(value).toBe(other)` assertions”

Jest Homepage





Standard Jasmine helpers (`beforeEach`, `afterEach`, etc.) are available

`describe()` defines a group of tests

`it()` defines a single test

Any failing assertions (`expect`, `assert`) will cause a test to fail

Tests will also fail if an error
any error is thrown

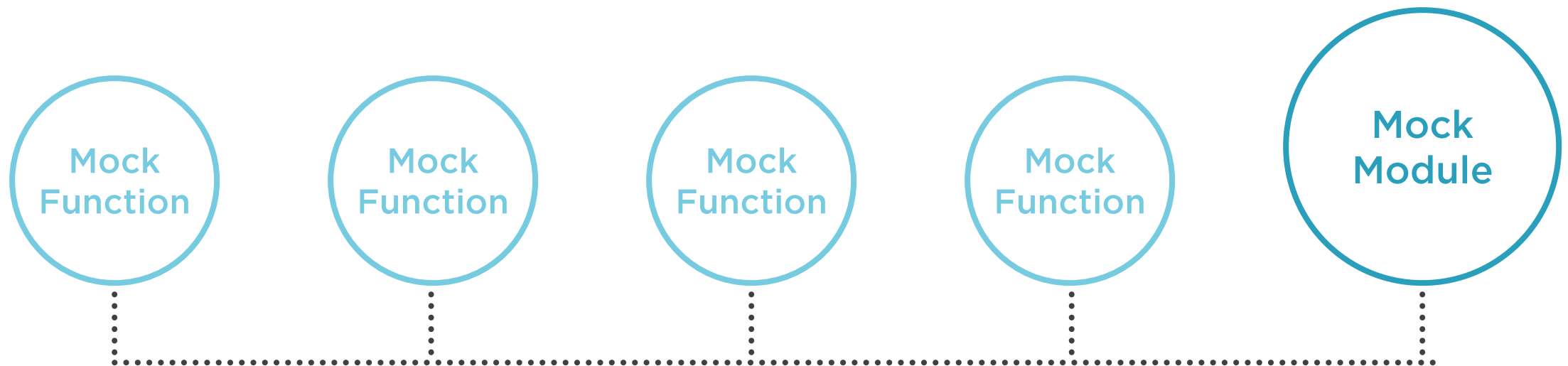


Mock Functions

Mock functions add functionality to mock modules

Mocked versions of modules don't have methods

Calling a non-existent function always results in error



Capabilities of Mocks

Functions can return
pre-determined values

Can verify how many times function
has been called; with what arguments

Can track objects instantiated from
a mock with the ``new`` keyword

Information contained within
the ``mock`` property of Jest mocks

Jest Jasmine has custom matchers,
used with mock property ->
``toBeCalled``, ``toBeCalledWith``

