TESTING 123

Tech Stack

- Test Runner and Framework
- Assertion Libraries
- Stubbing and Spying
- Testing React components

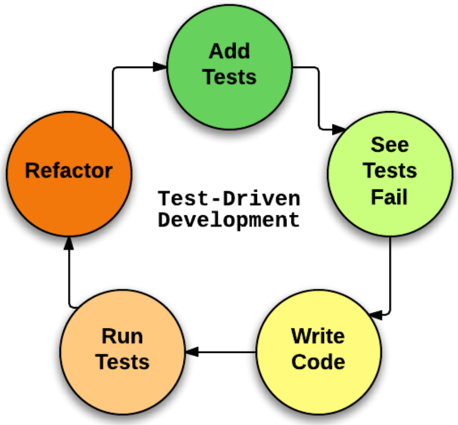
Test Runner and Framework

- **Karma - Runner**
  - Spins up real browsers
    - Presently we're using PhantomJS because we run headless browser tests on continuous integration servers.
  - Flexible configuration
    - Plugins that run with many testing frameworks like Mocha
    - JSPM config
    - Reporters
  - Doesn't actually run tests, it relies on a test framework for that such as Jasmine or Mocha.
- **Mocha - Framework**
  - Run against node.js or a browser
  - Write tests using the framework syntax of **describe** and **it** expressions
  - Before tests run - setup data, stub functions, and function spies using **before** and **beforeEach** expressions
  - After tests run - restore functions and stop spying using **after** and **afterEach** expressions (mainly so other tests aren't impacted by stubbed out functions)

Assertion Libraries

- **chai**
  - **assert, expect, should**
    - all 3 perform essentially the same task but in different ways
    - **assert** == Test Driven Development (TDD)
    - **expect** and **should** == Behavior Driven Development BDD

TDD



Assertion Libraries cont…

- chai-as-promised
    - improves the experience of testing promises

Stubbing and Spying

- Sinon
  - Stub functions to return override its return value
  - Spy on functions to ensure the they’re called and with the expected parameters
- sinon-chai
  - combines sinon and chai to write more elegant assertions on function calls

Testing React components

- Enzyme (airbnb)
    - Utilizes react-test-utils under the covers
        - this is great because react-test-utils are verbose and require more boilerplate
    - Shallow and full DOM rendering
    - Verify component props
    - Manipulate state
    - Simulate user interaction
    - Mimics jQuery’s API for DOM manipulation and traversal
    - Find elements and print HTML to run assertions on what is rendered

The End