

Integration Testing: You Keep
Using That Word, I Do Not Think
It Means What You Think It
Means

Melissa Marshall
@mveritym



Picture from www.red-badger.com

Case study

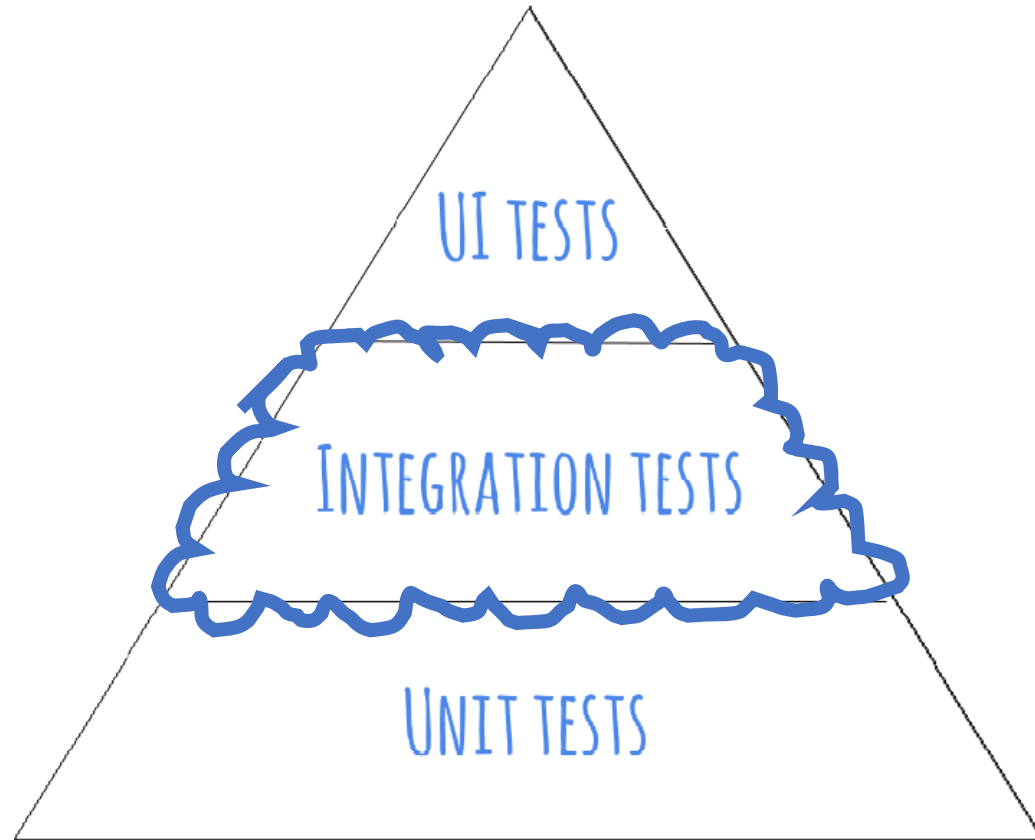
Unit test vs. Integration test



Source: Unknown via Giphy



Source: wendydoodles.tumblr.com via Giphy



What is integration testing?

- Unit testing: Jasmine, JUnit, OCUUnit, RSpec ...
- Functional testing: Protractor, Cucumber, Selenium ...
- Integration testing: ???

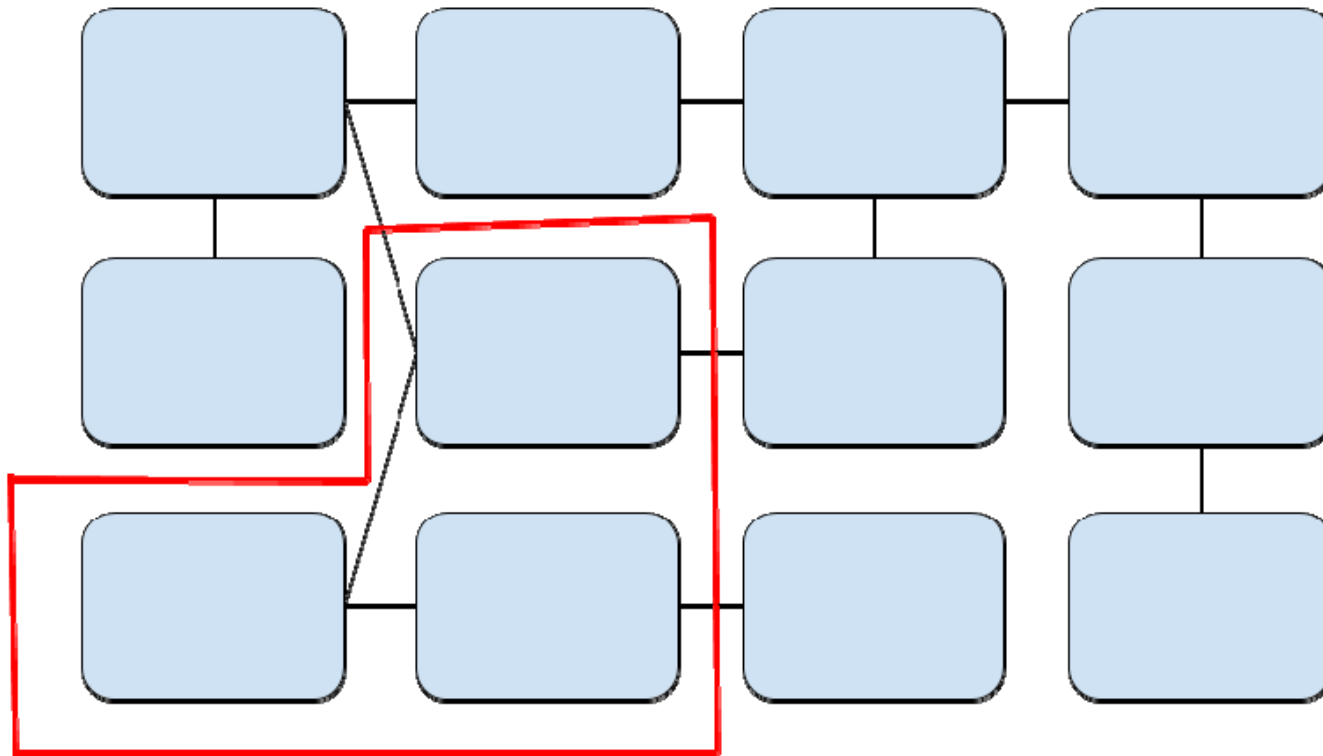
**Integration testing is controversial &
confusing**

→ used less often

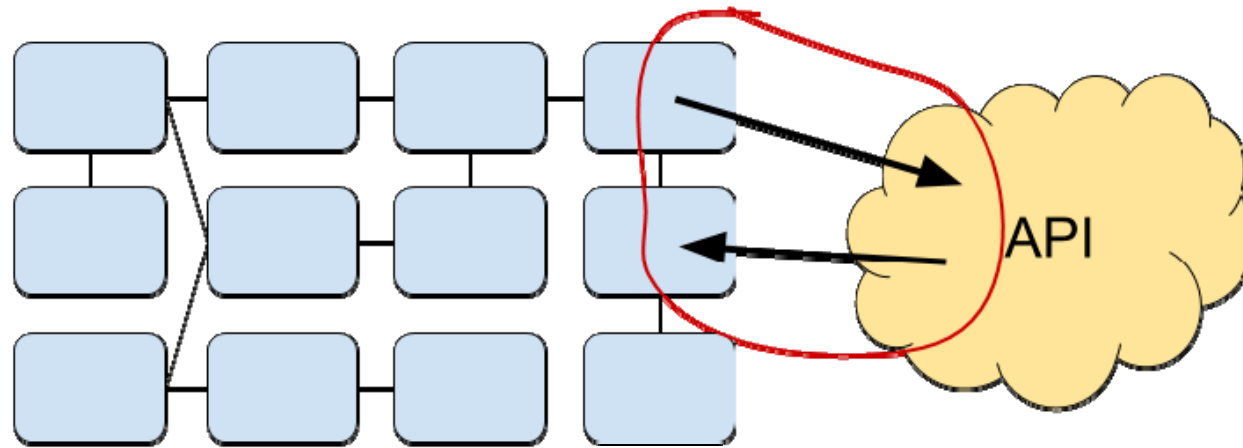
**There is no such thing as
“integration testing”.**

What you call “integration testing” depends on your tech and what you’re building.

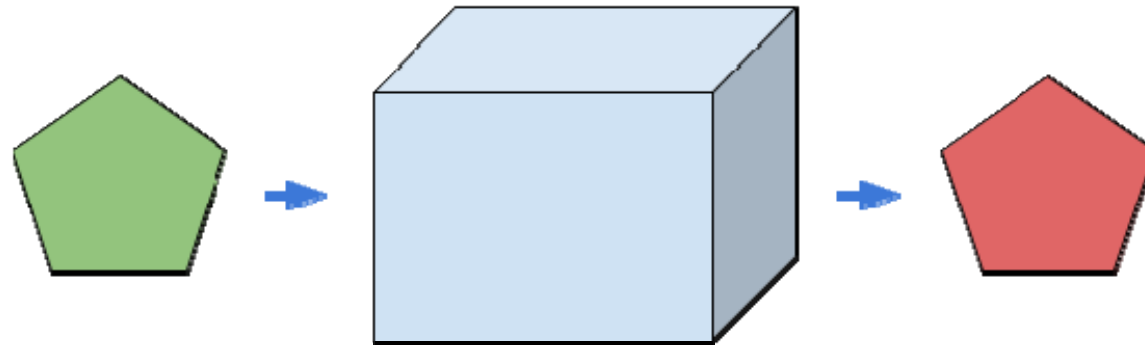
Example: group functions together, test them in aggregate



Example: test the boundaries of your applications

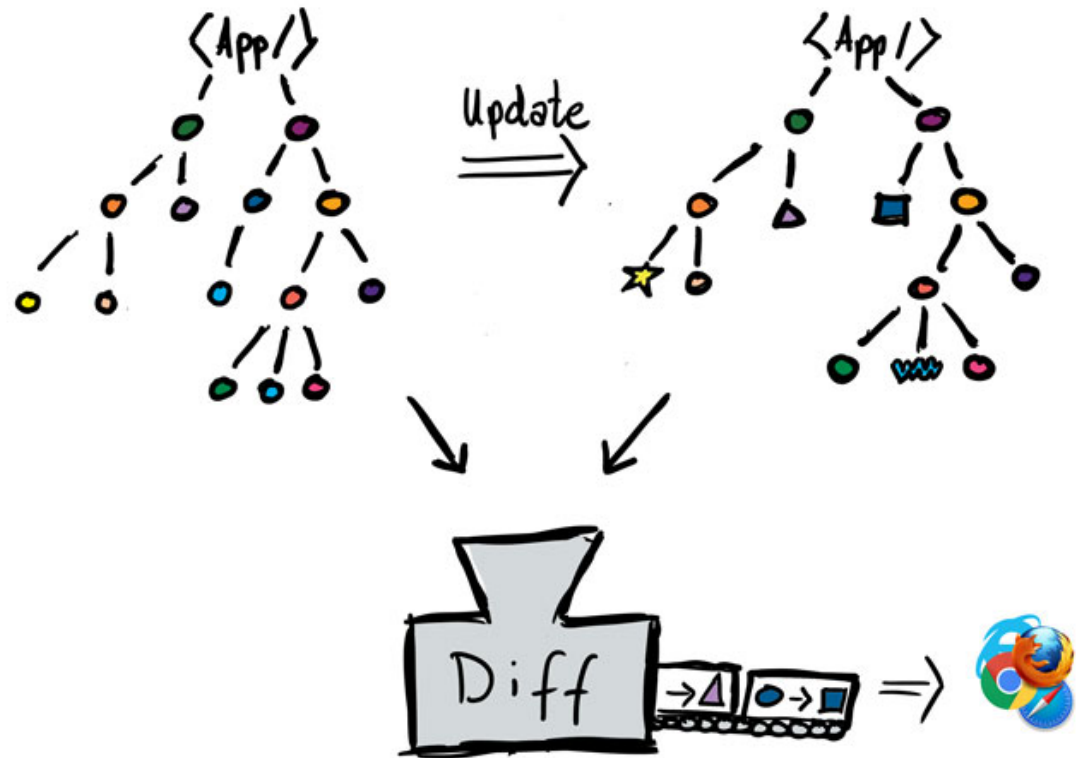


Example: use declarative UI or functional programming tools



Real World Example: testing a ReactJS app

- General idea:
put state in,
get DOM out



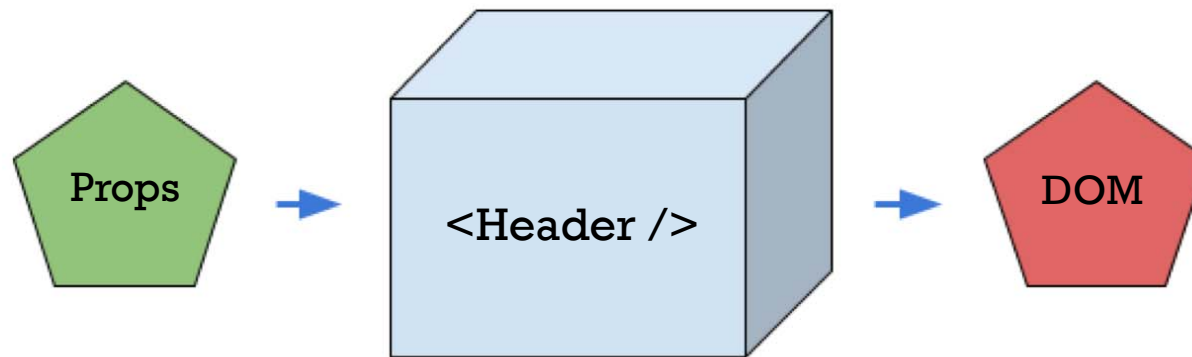
ReactJS

```
const Header = (props) => (  
  <div>  
    <h1>{props.title}</h1>  
    <h2>Written by {props.author}</h2>  
  </div>  
);
```

```
const App = () => (  
  <div id="container">  
    <Header title='My Article' author='Mel' />  
    <Article content='Wow what a great article' />  
    <Footer copyright='2017' />  
  </div>  
);
```

Enzyme

- Testing framework Enzyme has full DOM rendering
 - Render React component with initial state, inspect the DOM that's generated



```
import React from 'react';
import { mount } from 'enzyme';
import Header from './header.js';

describe('<Header />', () => {

  it('should render two header elements', () => {

    const props = {
      title: 'React JS Article';
      author: 'Melissa Marshall'
    };

    const wrapper = mount(<Header { ... props} />);

    expect(wrapper.html()).to.equal(`
      <div>
        <h1>React JS Article</h1>
        <h2>Written by Melissa Marshall</h2>
      </div>
    `);
  });
});
```



```
import React from 'react';
import { mount } from 'enzyme';
import Header from './header.js';

describe('<Header />', () => {

  it('should render two header elements', () => {

    const props = {
      title: 'React JS Article';
      author: 'Melissa Marshall'
    };

    const wrapper = mount(<Header { ...props} />);

    expect(wrapper.containsAllMatchingElements([
      <h1>React JS Article</h1>,
      <h2>Written by Melissa Marshall</h2>
    ])).to.equal(true);
  });
})
```

... a deterministic integration test!

- Testing a page without firing up phantomjs or using webdriver
- Generate test states from real scenarios
- FP type systems are reliable and deterministic.

The Zen of Testing



Source: [EmojiIsland](#)



Integration testing might not be the right thing for your team and that's totally OK.

Summary

- Integration testing is useful!
- ... but you need to figure out what you want from it
- Don't get too attached to tests
- Declarative UIs and functional programming are awesome

Thank You!

@mveritym