

Modern Web Testing: Going Beyond Selenium

Dmitry Vinnik, @DmitryVinnik

Lead Software Engineer, Salesforce

Goals

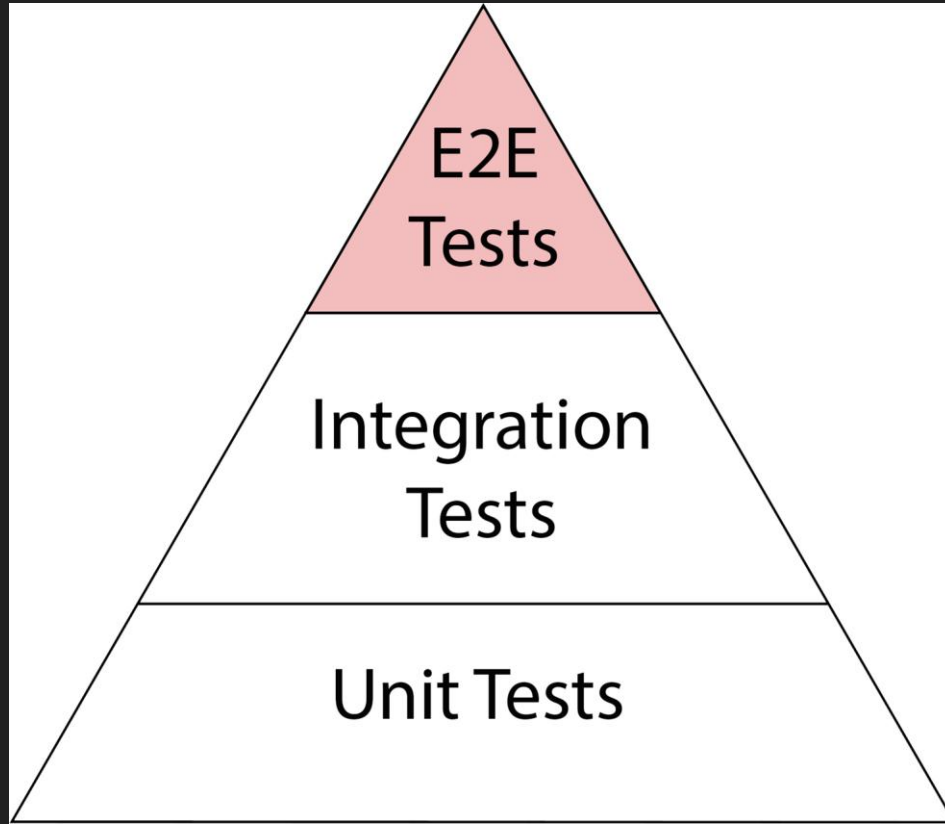
End-to-End Testing:
What, Why, How and When

Looking Beyond Selenium / VDT

Find Right E2E Tool For Your Team

How's doggie morality tests?

Test Pyramid



↓ Fidelity

↑ Speed

↓ Cost

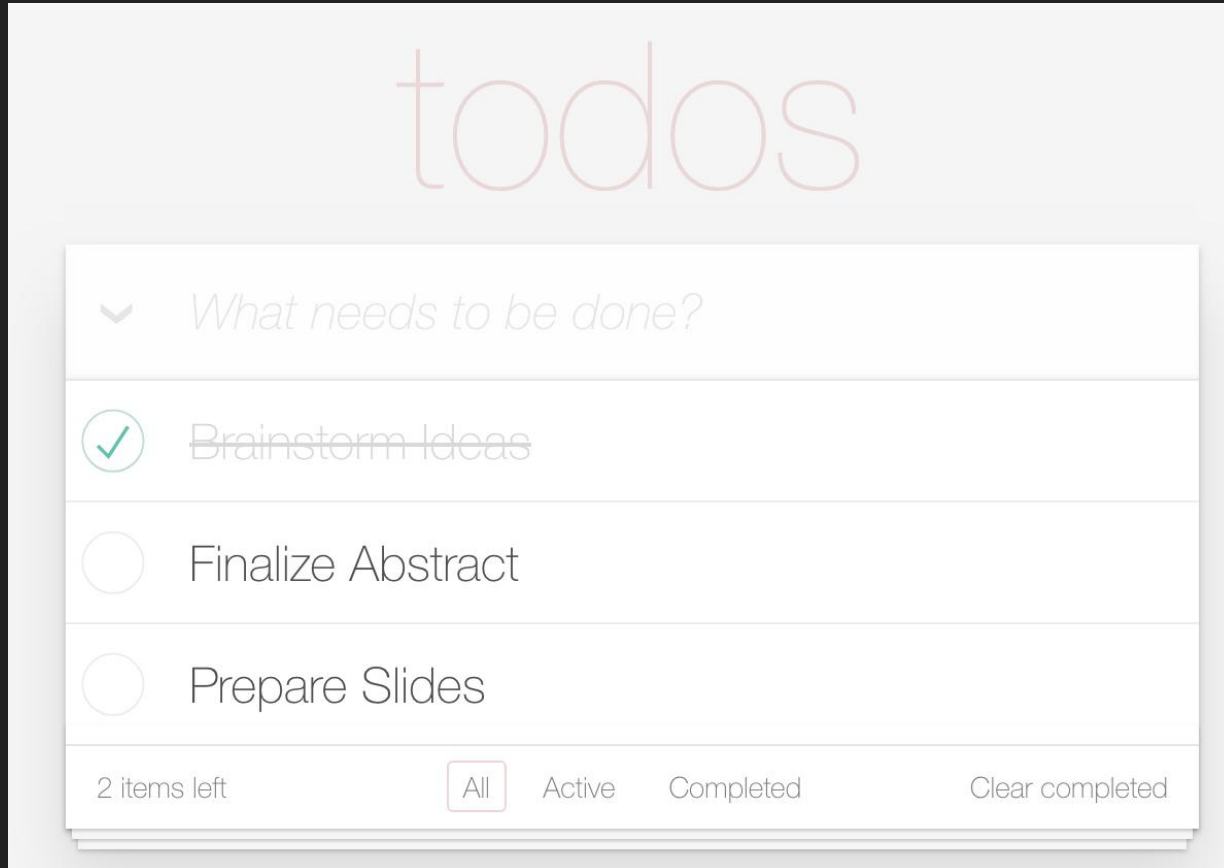
↑ Fidelity

↓ Speed

↑ Cost

Let's test something!
What is this “something”?

Example App



Back to testing!

End-to-End Testing Today's Focus

Common way to do E2E?

Defacto
E2E Tool

Testing
Standard

Selenium WebDriver

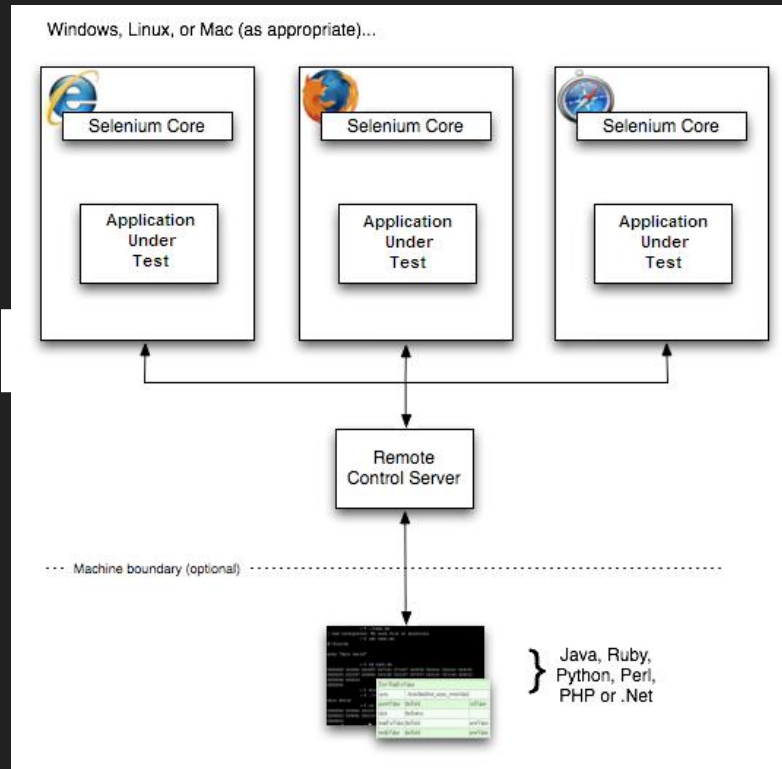
Generic

↑ Integrations

Selenium WebDriver

Sele

driver



From “SeleniumHQ, Selenium RC”

Backend:
Java

Case

Testing:
Java

Is there a problem?

Hybrid
Model

DDD

Why not Server Side Test?

User
Centric

Software
Engineers

Software
Principles

Hybrid Model

Testing
Expertise

Hybrid
Model

DDD

Why not Server Side Test?

User
Centric

**Ubiquitous
Language**

Domain Driven Design

Context Matters

Hybrid
Model

DDD

Why not Server Side Test?

User
Centric

Still not convinced?

Modern **Web** Testing

Jasmine

Cucumber

Power of JavaScript

Applertools

Speaking Language JavaScript of the Web Node.js

Isomorphic

Flexible

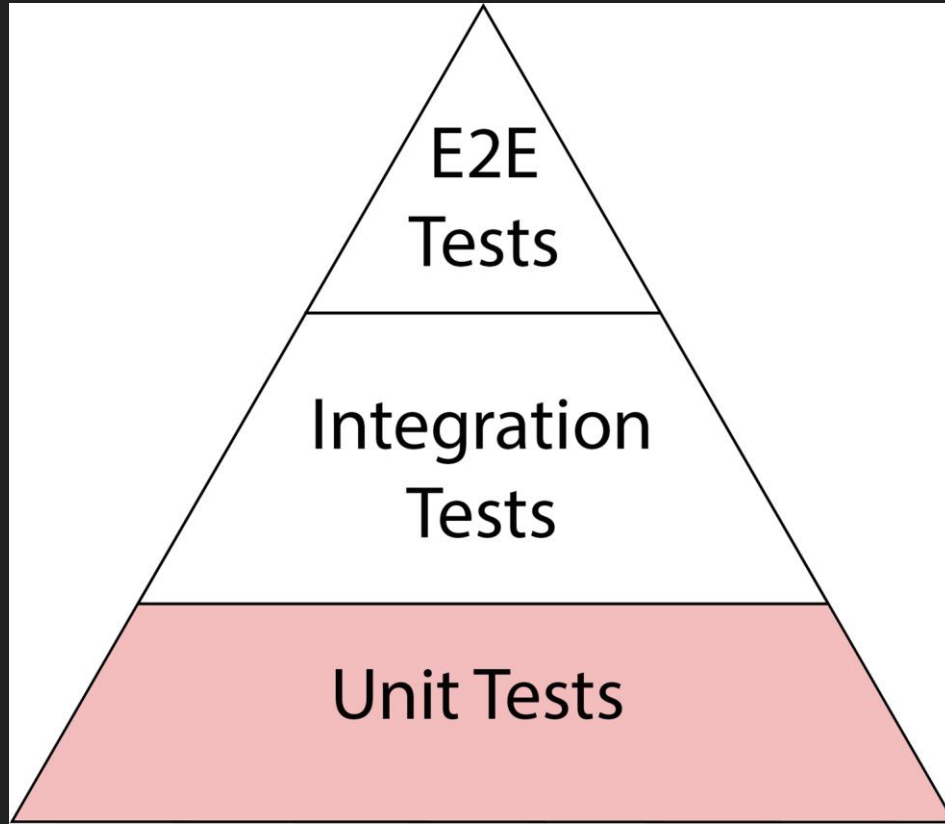
Node.js

Customizable

NPM
Manager

Go if established in a medium

Test Pyramid



↓ Fidelity

↑ Speed

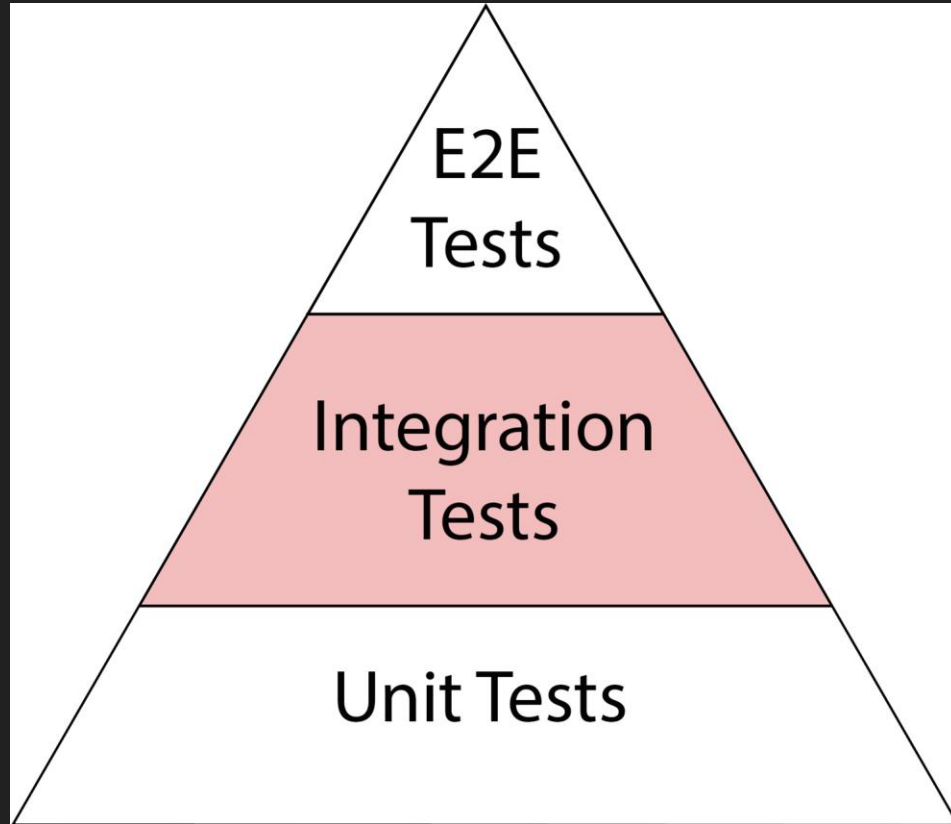
↓ Cost

Unit Testing In Web Context

Example: Jasmine

```
// Define Test Suite
describe("Create Todo Item", function () {
  // Define Test Case
  it("Accepts Valid Data", function () {
    var todoItem = createTodoItem('{ "name": "Brainstorm Ideas",
                                   "state": "incomplete"}');
    expect(todoItem.name).toEqual("Brainstorm Ideas");
    expect(todoItem.state).toEqual("incomplete");
  });
});
```

Test Pyramid



↓ Fidelity

↑ Speed

↓ Cost

Integration Testing In Web Context

Example: Jest

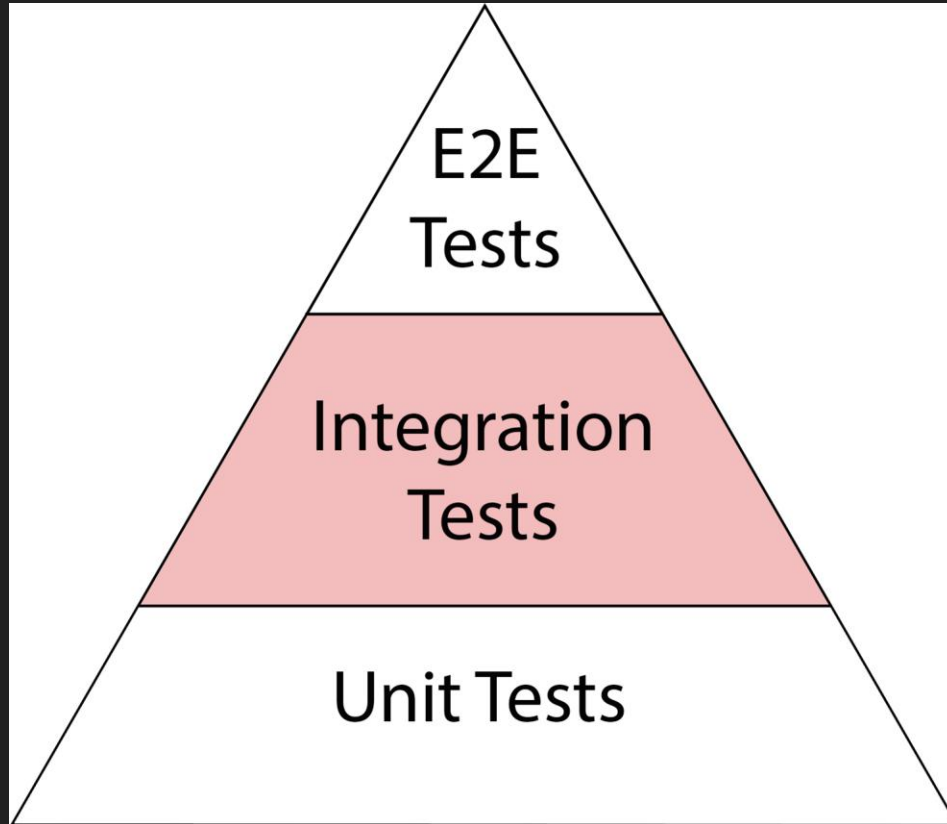
```
class CreateTodo {
  render() {
    return <button>
      createTodo
    </button>
  }
}

const todoStyle = {
  border: 'none',
  color: '#ffffff',
  padding: '15px 32px',
  fontSize: '17px',
  backgroundColor: '#555555',
};

export default class CreateTodo {
  render() {
    return <button
      className="createTodo"
      onClick={undefined}
      style={
        Object {
          "backgroundColor": "#555555",
          "border": "none",
          "color": "#ffffff",
          "display": "inlineBlock",
          "fontSize": "16px",
          "padding": "15px 32px",
          "textAlign": "center",
          "textDecoration": "none",
        }
      >
      createTodo
    </button>
  }
};
```

Let's reflect on it

Test Pyramid



↓ Fidelity

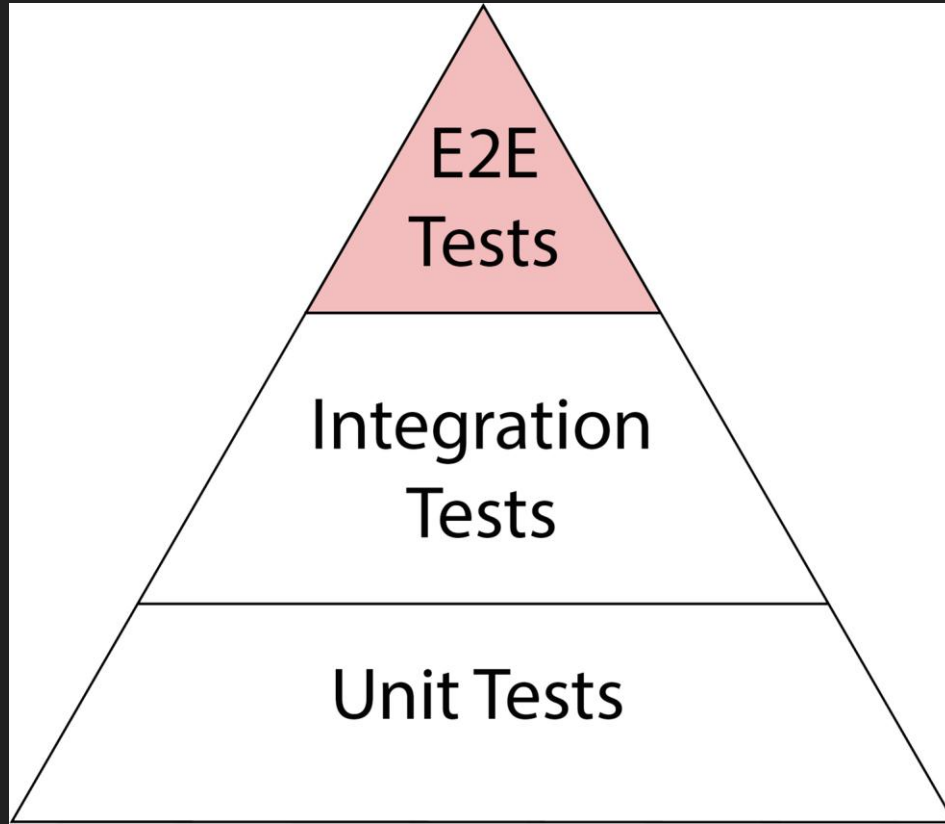
↑ Speed

↓ Cost

Every step has
Developers
only 1 focus

Bad for environment

Test Pyramid



↓ Fidelity

↑ Speed

↓ Cost

↑ Fidelity

↓ Speed

↑ Cost



But should we read it?

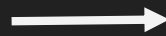
~~Interactions~~



Tasks

~~User-Centered Task Testing~~

~~Page Object Model~~



Screenplay Pattern

Should we use
Back for our app
Java for tests?

Wartest du die mit der Seite?

Continuum with Selenium

Defacto
E2E Tool

Testing
Standard

SeleMVerndrAverDsiver

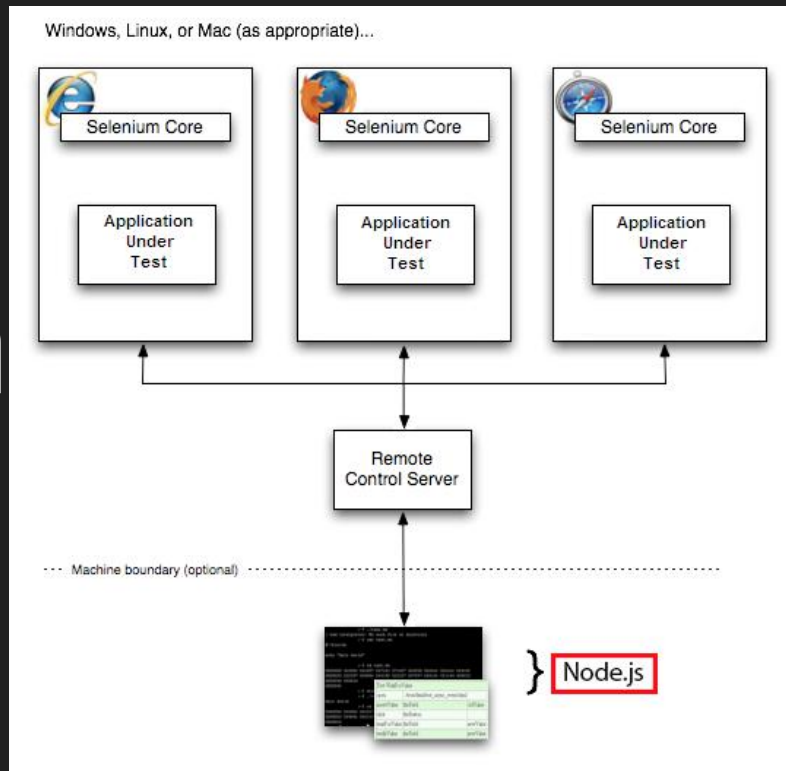
Generic

↑ Integrations

Selenium WebDriverJs

Selen

iverJs



From “SeleniumHQ, Selenium RC”

Example: WebDriverJs

```
suite(function(env) {
  describe('Create Todo', function() {
    let driver;
    before(async function() {
      driver = await env.builder().build();
    });

    it('Using Valid Data', async function() {
      await driver.get('http://localhost:3000/');
      await driver.findElement(By.className('todoName'))
        .sendKeys('Brainstorm Ideas', Key.RETURN);
      await driver.wait(until.elementTextIs(driver
        .findElement(By.className('todoItem')), 'Brainstorm Ideas'), 1000);
    });

    after(() => driver && driver.quit());
  });
});
```

Only good deeds, right?

Selectors

Locator
Strategy

E2E Tests:

General Complexity

Test Flows

SPA Types

What about WebDriverJs?

ID, CSS,
Xpath

By Locators

E2E Tests:
WebdriverJs
General Complexity

Promise
Manager

Generic

Is the ~~leak~~ ~~key~~ thing ~~is~~ ~~the~~ ~~letter~~?

Models

Controllers

Remember AngularJS test app?

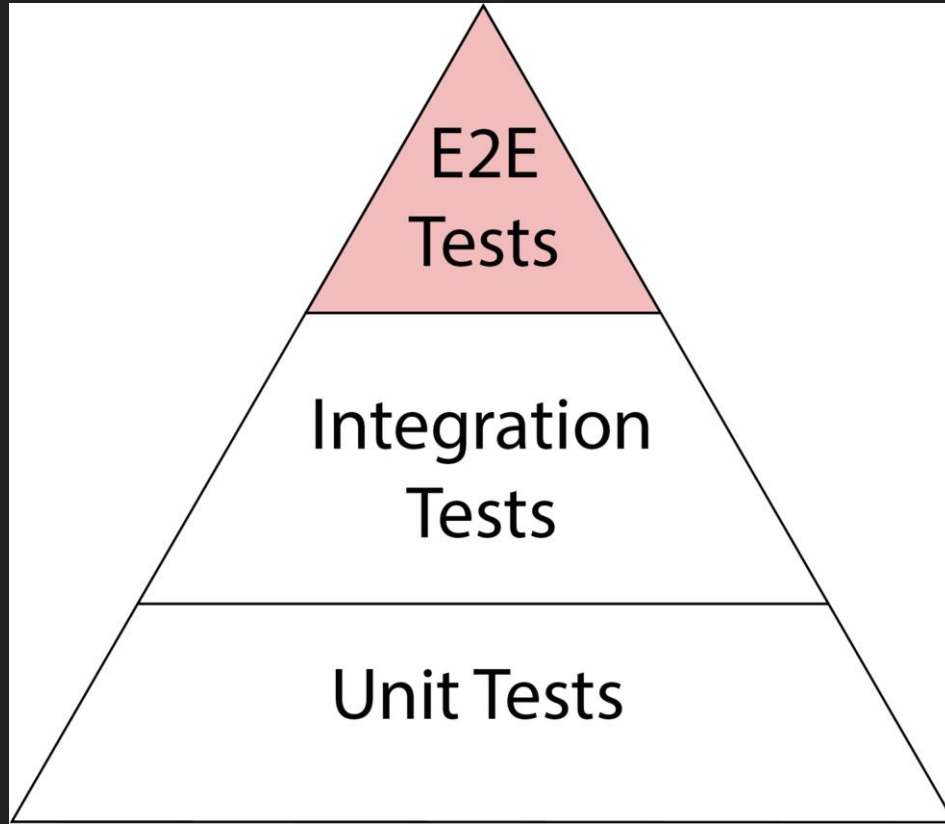
Components

Demo App: Closer Look

```
<form id="todo-form" ng-submit="addTodo()" />
<input class="toggle" type="checkbox"
      ng-model="todo.completed"
      ng-change="toggleCompleted(todo)">
<a ng-class="{selected: status == '}'
<ul id="todo-list">
  <li ng-repeat="todo in todos"
      ng-class="{completed: todo.completed}" />
</ul>
```

How would we test that?

Test Pyramid



↓ Fidelity

↑ Speed

↓ Cost

↑ Fidelity

↓ Speed

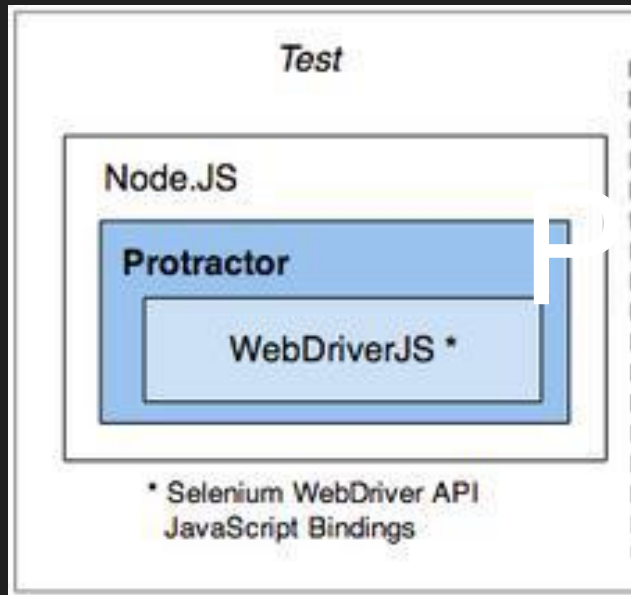
↑ Cost

Do we have to use
No!
WebdriverJs?

Protractor

E2E Test Framework for Angular Apps

Protractor



From “Protractor.org”

Element
Bindings

By Repeaters

E2E Tests:
Protractor
General Complexity

Application
State Aware

Angular
Oriented

Example: Protractor

```
describe('Create Todo', function() {
  it('Using Valid Data', function() {
    browser.get('http://localhost:3000/');
    element(by.model('newTodo'))
      .sendKeys('Brainstorm Ideas', Key.RETURN);
    expect(element(by.binding('todo'))
      .getText()).toEqual('Brainstorm Ideas');
  });
});
```

Angular is great

React

Vue

There are Many More Front-
End Frameworks...

Ember

Backbone

Protractor + React

```
onPrepare: function() {  
    browser.ignoreSynchronization = true;  
}
```

~~Production React~~

TestCafe

Non-Selenium What's Next? UI Test Frameworks

Cypress.io

Handling
Waits

Parallel
Execution

What patterns we
Patterns everywhere

RTD

Recorder
& IDE

TestCafe

Non-Selenium UI Test Frameworks

Cypress.io

Handling
Waits

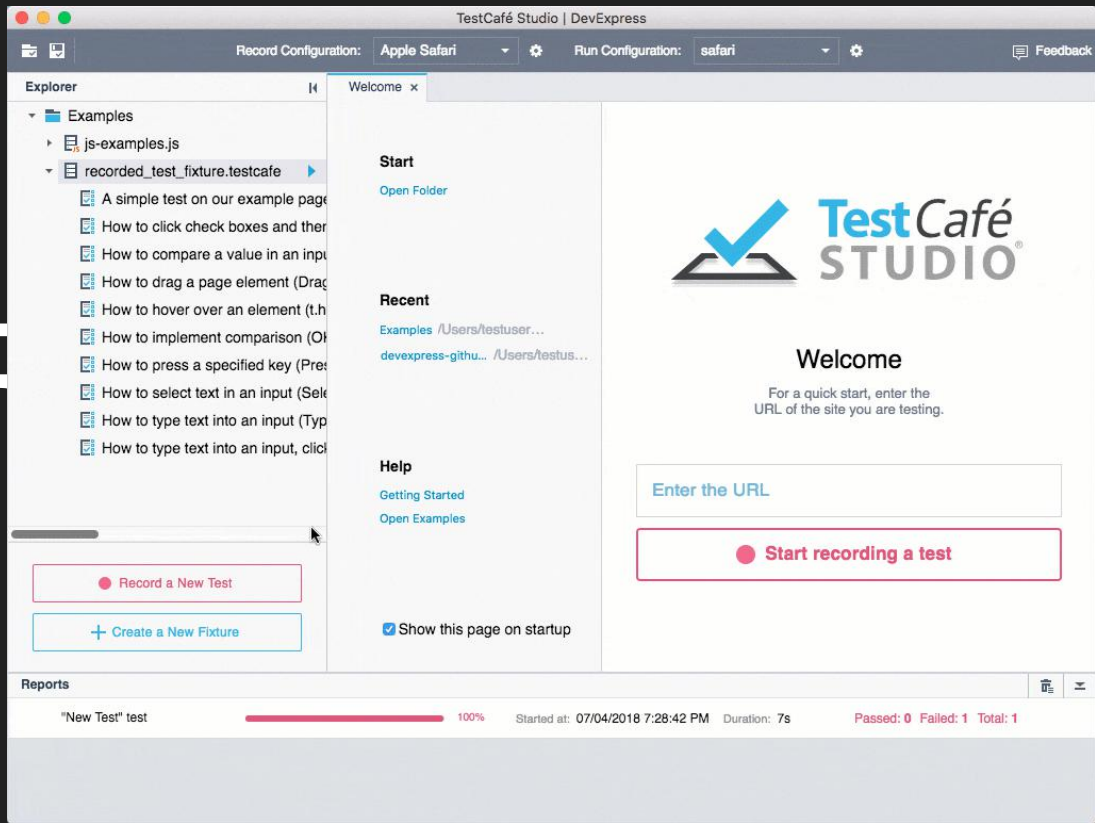
Parallel
Execution

Common Patterns

RTD

Recorder
& IDE

TestCafe: Recorder & IDE



Source:

TestCafe by DevExpress, <https://github.com/DevExpress/testcafe>

TestCafe: Example

```
import { Selector } from 'testcafe';

fixture `Getting Started`
  .page `https://devexpress.github.io/testcafe/example`;

test('My first test', async testCase => {
  await testCase
    .typeText('#developer-name', 'John Smith')
    .click('#submit-button')

    .expect(Selector('#article-header').innerText).eql('Thank you!');
});
```

Source:

TestCafe by DevExpress, <https://github.com/DevExpress/testcafe>

TestCafe

Non-Selenium UI Test Frameworks

Cypress.io

Handling
Waits

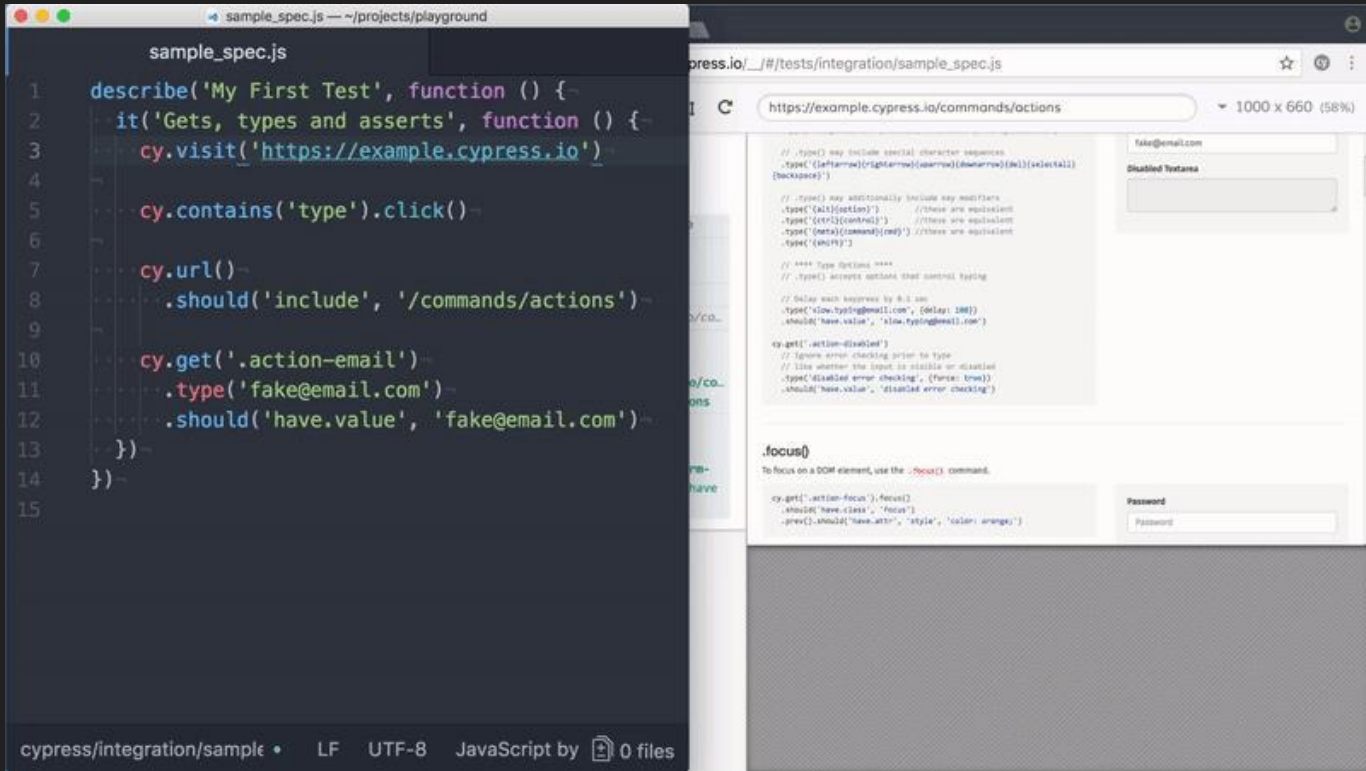
Parallel
Execution

Common Patterns

RTD

Recorder
& IDE

Cypress.io: Reports & IDE



Source:

Cypress.io Example by Cypress.io, <https://www.cypress.io>

Cypress.io: Example

```
describe('The Login Page', function () {
  it('logging in via form submission', function () {
    cy.visit('http://localhost:3000/login')
    cy.get('input[name=username]').type(username)
    cy.get('input[name=password]').type(`${password}{enter}`)

    cy.url().should('include', '/loggedInScreen')
    cy.getCookie('your-session-cookie').should('exist')
    cy.get('h1').should('contain', 'dvinnik')
  })
})
```

Source:

Basic Example by Cypress.io, <https://www.cypress.io>

Right Tool for the Right Problem

WebdriverIO

Nightwatch.js

There are many other
E2E Test Frameworks

WD

Nemo

↑ Complexity

No DDD

Avoid Tool Mixups

No Common
Language

**Github
Stars**

**NPM
Downloads**

How to Choose?

**External
Integrations**

POC

Flexibility

But More Importantly

Use Case

Cost of
Transition

ROI

Flexibility

Ability to
Customize

Ability to
Replace

Flexibility

But More Importantly

Use Case

Team
Expertise

Application
Framework

Use Case

Test
Infrastructure

Evaluate Test
Architecture

Have Domain
Boundaries

Call For Action

Unify Test
Strategy

Experiment!

Q/A

About Speaker

Twitter: @DmitryVinnik

LinkedIn: in/dmitry-vinnik/

Email: dmitry@dvinnik.com