React Alicante

End to End testing React applications

Forbes Lindesay

September 2017

Thanks to our sponsors!

balsamiq®

socialpoint

stackoverflow en español

Brains & BEARDS

flywire

Limenius

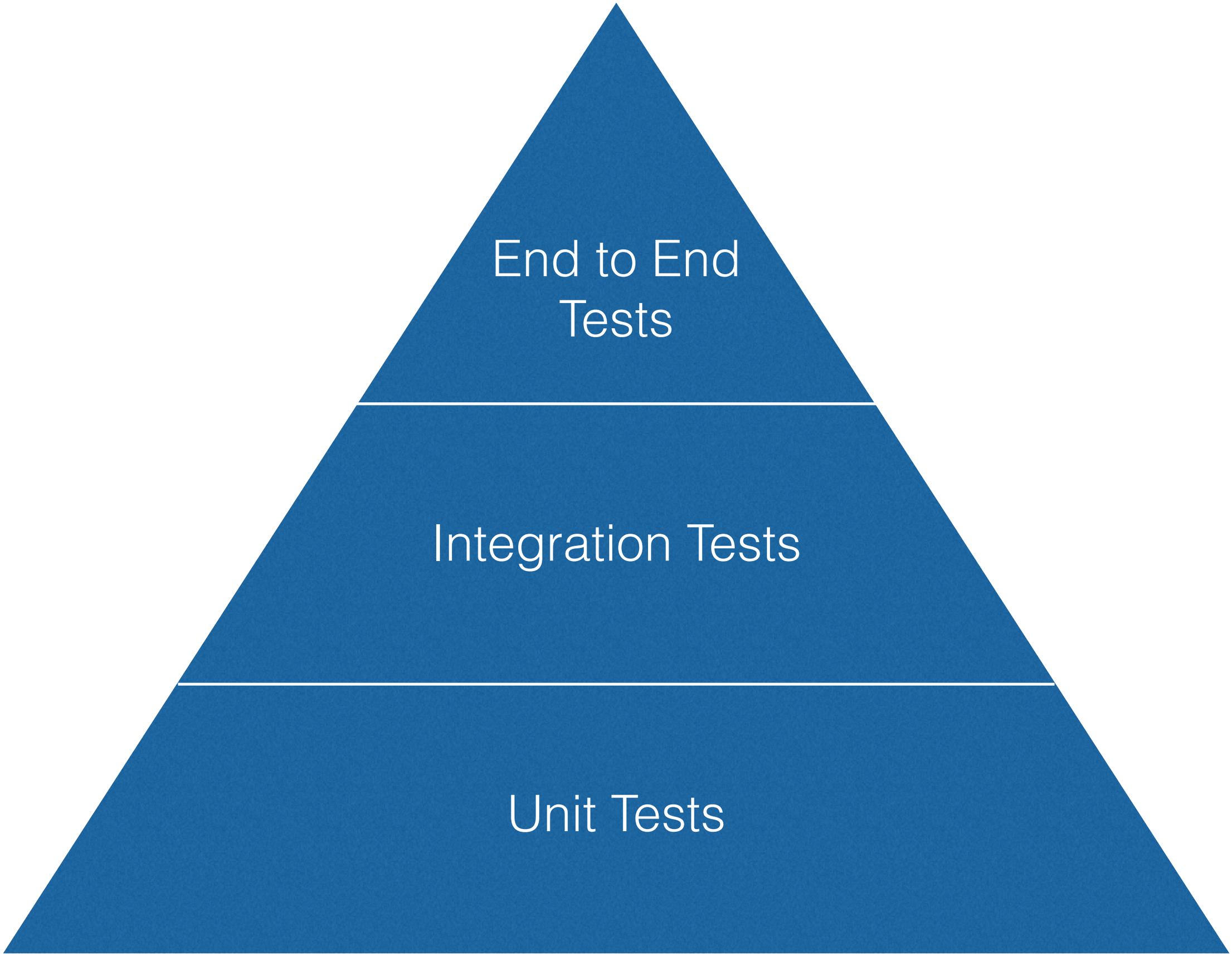
lemoncode

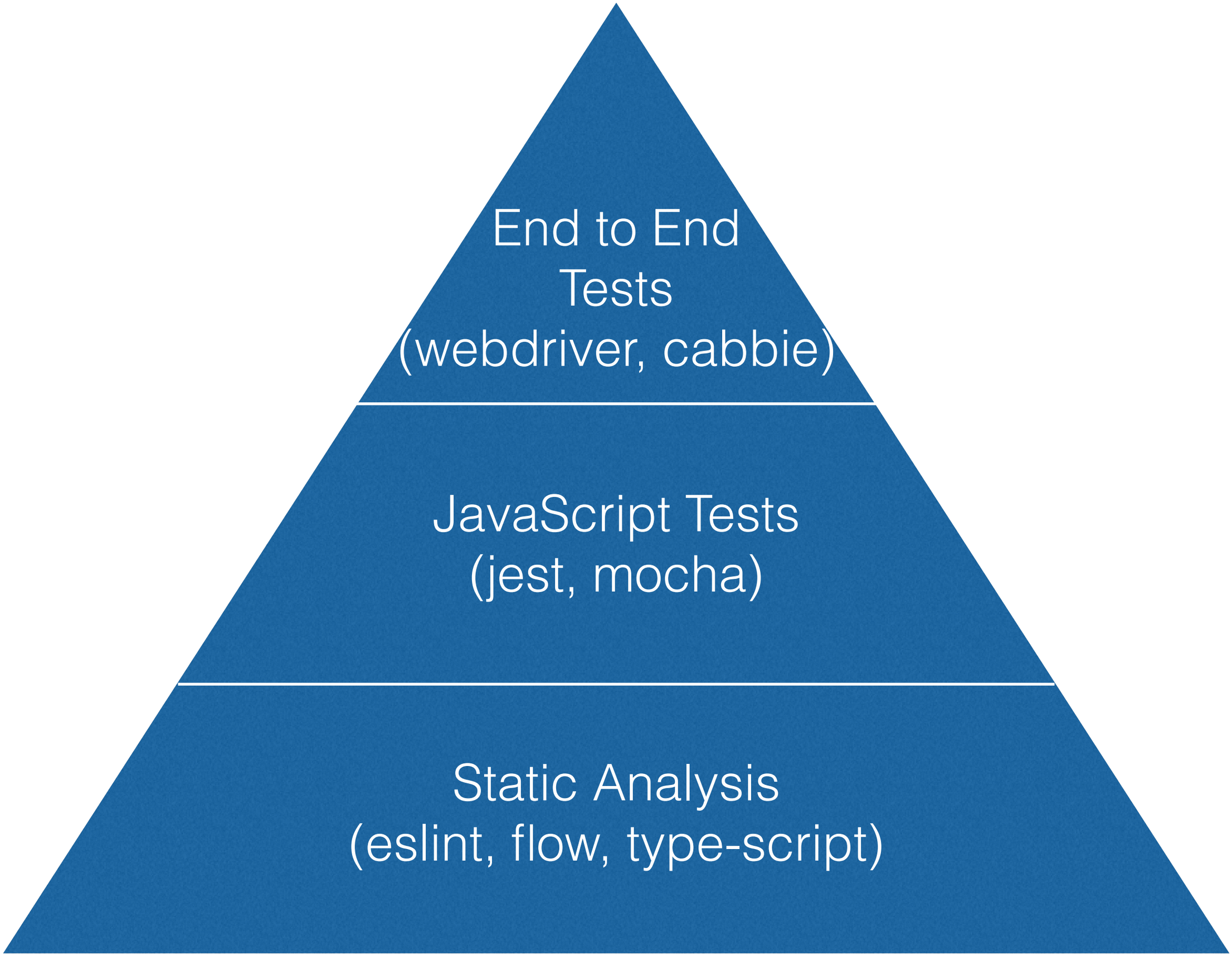
NITSNETS

React Alicante

NativeBase by GeekyAnts

ULab IDEAS MEETING POINT


End to End Tests
Integration Tests
Unit Tests


End to End
Tests
(webdriver, cabbie)
JavaScript Tests
(jest, mocha)
Static Analysis
(eslint, flow, type-script)

End to End
Tests
(webdriver, cabbie)

JavaScript Tests
(jest, mocha)

Static Analysis
(eslint, flow, type-script)

Why:
Static Analysis

- Cover 100% of code
- Built in documentation (sort of)
- Catches incorrect assumptions (e.g. numbers represented as strings)
- Help make refactors safe

Limitations of: Static Analysis

- Doesn’t encode **intent**
- Doesn’t actually run the code

How To:
Static Analysis

- Eslint / Tslint - Run rules to check for mistakes
- Typescript - Static type checking extensions to JavaScript
- Flow - Static type checking extensions to JavaScript

prettier

Why:
JavaScript Tests

- Much faster than end to end tests
- Reliable/Consistent
- Isolates problems

Limitations of: JavaScript Tests

- Doesn’t test the actual platform
- You need a lot of tests to get good coverage
- Doesn’t test integration between frontend and backend
- Doesn’t test what a real user would do

JEST

Why:
End to End Tests

- Test what the real user would do
- Expose browser inconsistencies
- A relatively small number of tests can cover the integration of huge parts of your system
- The only way to **know** a part of your system is working

Limitations of: End to End Tests

- Very slow
- Unreliable
- Expensive
- It’s hard to know what caused failures

Webdriver

WebDriver is a remote control interface that enables introspection and control of user agents. It provides a platform- and language-neutral wire protocol as a way for out-of-process programs to remotely instruct the behavior of web browsers.

Webdriver

a consistent API for automating browsers

https://w3c.github.io/webdriver/webdriver-spec.html

WebDriver

W3C Candidate Recommendation 26 September 2017

This version:

https://www.w3.org/TR/2017/CR-webdriver-20170926/

Latest published version:

https://www.w3.org/TR/webdriver/

Latest editor's draft:

https://w3c.github.io/webdriver/webdriver-spec.html

Implementation report:

https://github.com/w3c/webdriver/blob/master/implementation-report.md

Previous version:

https://www.w3.org/TR/2017/WD-webdriver-20170329/

Editors:

Simon Stewart

David Burns, Mozilla

Participate:

GitHub w3c/webdriver

Open bugs

#webdriver on irc.w3.org

Cloud Providers

SAUCELABS

BrowserStack

TestingBot

CrossBrowserTesting
A SMARTBEAR COMPANY

JSDOM

JSDOM + TAXI-RANK

Cabbie

Demo Time

canoeslalomentries — -bash — 147×43

~/GitHub/canoeslalomentries — -bash | ~/Downloads — -bash | node · npm MANPATH=/Users/...C=https://lcjs.org/dist/index.tab ...

```
Forbess-MacBook-Air-5580:canoeslalomentries forbeslindesay$
```

Demo Time

Identifying Elements

- Give them a “data-test-id” attribute
- Do not rely on position within the document

Handling delays

- Never just add a timeout to your test
- Poll the website until it meets the condition

Complex Elements

```
componentDidMount() {
  if (this.props['data-test-id']) {
    window.selects = window.selects || {};
    window.selects[this.props['data-test-id']] = this;
  }
}
componentWillUnmount() {
  if (
    this.props['data-test-id'] &&
    window.selects &&
    window.selects[this.props['data-test-id']]
  ) {
    delete window.selects[this.props['data-test-id']];
  }
}
```

Complex Elements

```
export function getSelectValue(dataTestID: string) {
  return driver.activeWindow.execute(
    'window.selects[arguments[0]].props.value',
    [dataTestID],
  );
}
export function setSelectValue(dataTestID: string, value: any) {
  driver.activeWindow.execute(
    'window.selects[arguments[0]].props.onChange(arguments[1])',
    [dataTestID, value],
  );
}
```

Live Demo Time

Thank You