



Make backend devs jealous of your test suite

Carly Litchfield
January 25, 2018

Why Test

How to Test

Conclusion

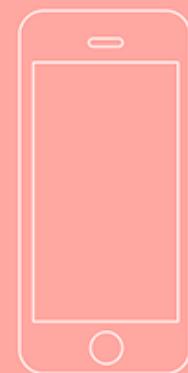
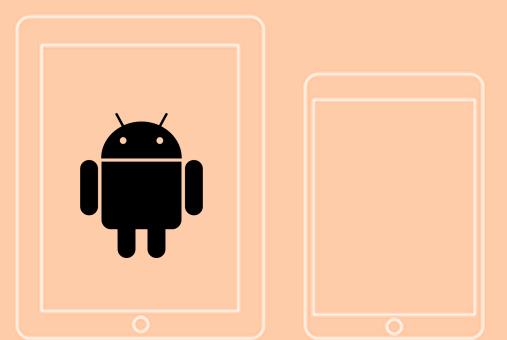
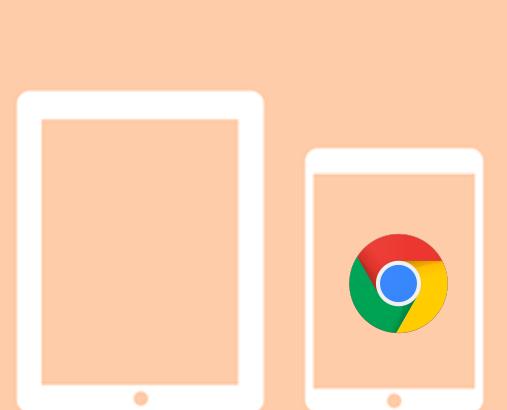
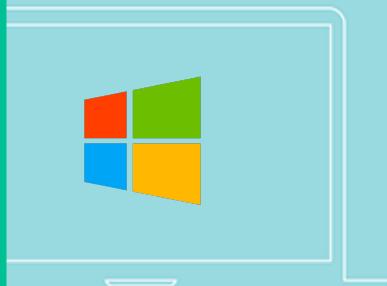
Backend devs writing frontend code like



1. CSS

```
width: 100% !important;
height: 100%;
background: rgba(0,0,0,.75);
transition: opacity 300ms;
z-index: 500 !important;
}#overlay.hide {
pointer-events: none;
opacity: 0;
}.confirmation .actions {
border-top: 1px solid #eee;
padding: 5px !important;
text-align: right;
background: #fafafa;
#dialog-scale-hide {
#dialog-scale-hide inset 0 1px 0 white;
-webkit-transform: scale(0);
-moz-transform: scale(0);
border: 1px solid #888;
#overlay {
position: fixed;
top: 0;
left: 0;
background: #fafafa;
z-index: 500 !important;
width: 100% !important;
background: rgba(0,0,0,0.75);
transition: opacity 300ms;
}.color-picker canvas {
border: 1px solid #eee;
background: #fff;
width: 100% !important;
height: 100%;
pointer-events: none;
}#overlay {
position: fixed;
top: 0;
left: 0;
opacity: 1;
width: 100% !important;
height: 100% !important;
background: #fff;
border: 1px solid #888;
background: rgba(0,0,0,0.75);
transition: opacity 300ms;
}.confirmation .actions {
z-index: 500 !important;
top: 1px solid #eee;
}#overlay.hide {
padding: 5px !important;
pointer-events: none;
text-align: right;
opacity: 0;
background: #fafafa;
}.confirmation .actions {
border: 1px solid #eee;
background: #fff;
width: 100% !important;
height: 100% !important;
border-top: 1px solid #888;
padding: 5px !important;
text-align: right;
background: #fafafa;
}#overlay.hide {
background: #fafafa;
```

1. CSS
2. Hard to tell if they broke anything



DON'T NEED TESTS



IF YOU DON'T WRITE BUGS

More than catching bugs

Save a dev, write a test

Better than any readme

```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```

Signal to refactor

```
55 [SetUp]
56 public void SetUp()
57 {
58     // Set up mocks.
59     _mRepo = _mocks.Mock<ISpecialtyProcedureMappingRepository>();
60     Func<ICoordinator, ISpecialtyProcedureMappingRepository> repoFactory = coord => _mRepo.Object;
61     _mocks.Provide(repoFactory);
62     _mProfessionalCloudUpdateService = new Mock<ICloudUpdateService>();
63     var lazyCloudUpdateServiceMock = new Lazy<ICloudUpdateService>(() => _mProfessionalCloudUpdateService.Object);
64     _mocks.Provide(lazyCloudUpdateServiceMock);
65
66     // By decree: specialty #N uses procedure #N, for all N.
67     _procIdsBySpecId = specId => new List<long> { specId };
68     _specIdsByProcId = procId => new List<long> { procId };
69
70     _mockProfessionals = new Dictionary<long, Mock<IProfessional>>();
71     _mockSpecialties = new Dictionary<long, Mock<ISpecialty>>();
72     _mockProcedures = new Dictionary<long, Mock<IProcedure>>();
73
74     _mSpecialtyRepo = _mocks.Mock<ISpecialtyRepository>();
75     _mSpecialtyRepo.Setup(x => x.GetSpecialty(It.IsAny<long>()))
76         .Returns((long s) => _mockSpecialties[s].Object);
77     _mSpecialtyRepo.Setup(x => x.GetSpecialtyIfNotDeleted(It.IsAny<long>()))
78         .Returns((long s) =>
79             _specialtyIdsThatAreDeleted.Contains(s) ? null : _mockSpecialties[s].Object);
80     _mSpecialtyRepo.Setup(x => x.GetProcedure(It.IsAny<long>()))
81         .Returns((long s) => _mockProcedures[s].Object);
82     _mSpecialtyRepo.Setup(x => x.GetSpecIdsByProcId(It.IsAny<long>())).Returns((long p) => _specIdsByProcId(p));
83     _mSpecialtyRepo.Setup(x => x.GetDoctorAvailableProceduresBySpecId(It.IsAny<long>()))
84         .Returns((long s) =>
85             _procIdsBySpecId(s).Select(p => GetOrCreateMockProcedure(p).Object).ToList());
86
87     _mUserRepo = _mocks.Mock<IUserRepository>();
88     _mUser = _mocks.Mock<IZDMembershipUser>();
89     _mProviderApprovalRequirementsService = _mocks.Mock<IProviderApprovalRequirementsService>();
90
91     // Set up shared 'backing' state.
92     _specialtyIdsThatAreDeleted = new HashSet<long>();
93     _procedureIdsWithNoDoctorAvailable = new HashSet<long>();
94     _resourceMappings = new Dictionary<long, ISet<long>>();
95     _professionalIdsThatCannotBookDiagnostics = new HashSet<long>();
96     _professionalIdsWithProfileCompleteFirstCall = new HashSet<long>();
97     _professionalIdsWithProfileCompleteSecondCall = new HashSet<long>();
98     _professionalIdsReturningNewApplicationTrue = new HashSet<long>();
99     _professionalsForResourcesProvider = new Dictionary<long, ISet<long>>();
100    _professionalSubStatus = new Dictionary<long, SubStatus>();
101
102    // Set up to send events to cloud
```

Tests



Continuous Integration

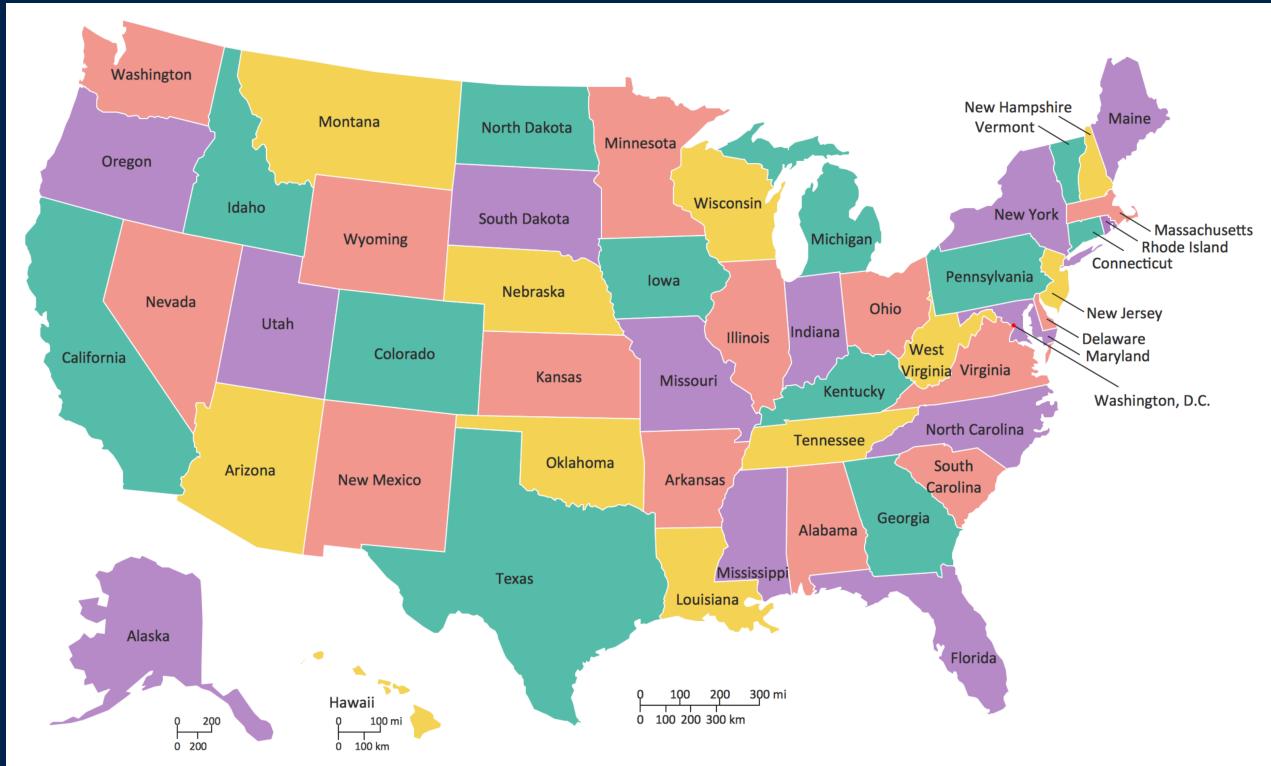


Continuous Deployment



Beat the Competition

Why Test
How to Test
Conclusion



America as seen by Europeans

CALIFORNIA

TEXAS

NEW YORK



COWBOY HOROSCOPE

UNLOCK THE WISDOM OF THE STARS, FARMERS, AND THE GREAT AMERICAN WEST.

GET YOUR COWBOY HOROSCOPE NOW

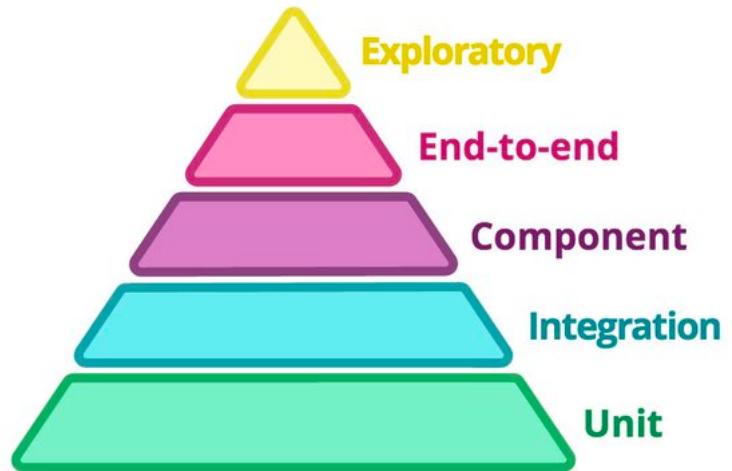
Enter the month and day of your birthday

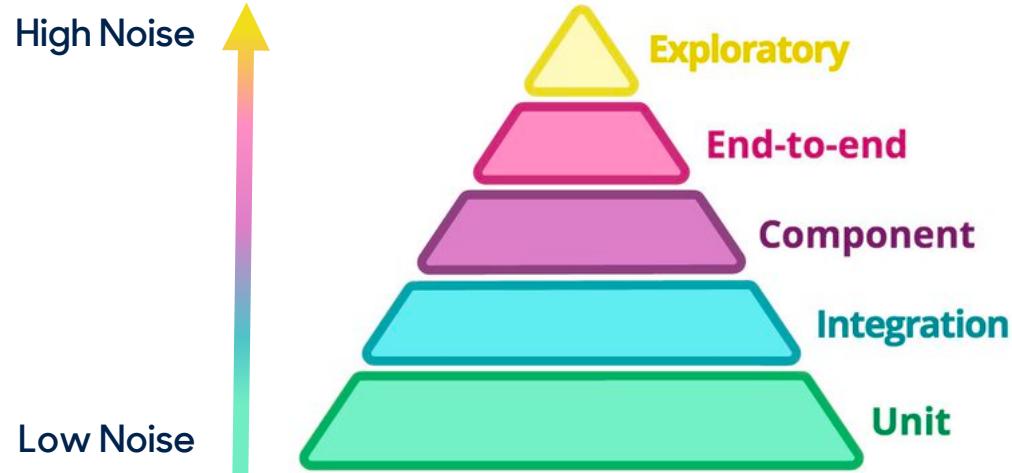
MONTH

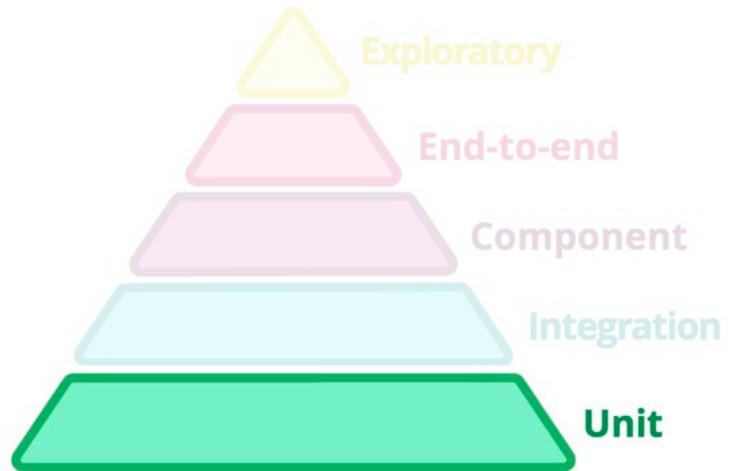
DAY

REVEAL YOUR COWBOY HOROSCOPE











Jest

```
it('does a thing', () => {
  assume();
  act();
  assert();
});
```

Enter the month and day of your birthday

MONTH

DAY

REVEAL YOUR COWBOY HOROSCOPE

Enter the month and day of your birthday

MONTH

Select... ▾

DAY

Select... ▾

Third Party
Component

REVEAL YOUR COWBOY HOROSCOPE

Enzyme

```
it('has two BirthdaySelects', () => {
  const wrapper = shallow(<BirthdayForm />);
  expect(wrapper.find(BirthdaySelect)).toHaveLength(2);
});
```

```
it('sets the month class on the icon', () => {
  const wrapper = mount(
    <BirthdayForm />
  );
  // simulate month selection
  wrapper.find('input').at(0).simulate('change',
    { target: { value: 'March' } }
  );
  // simulate day selection
  wrapper.find('input').at(1).simulate('change',
    { target: { value: '12' } }
  );
  const monthIcon = wrapper.find(Icon).at(0);
  expect(monthSelect.props().className).toEqual('pisces');
});
```

```
it('sets the month class on the icon', () => {
  const wrapper = mount(
    <BirthdayForm />
  );
  // simulate month selection
  wrapper.find('input').at(0).simulate('change',
    { target: { value: '12' } }
  );
  // simulate day selection
  wrapper.find('input').at(1).simulate('change',
    { target: { value: '12' } }
  );
  const monthIcon = wrapper.find(Icon).at(0);
  expect(monthSelect.props().className).toEqual('pisces');
});
```



A brief plug for Redux

***you might not need Redux**



Business Logic

reducers, actions,
`mapStateToProps`, etc...

Markup

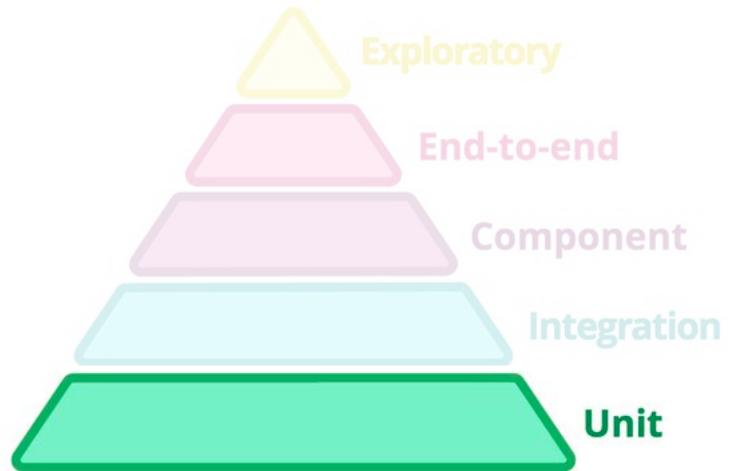
... `extends Component`

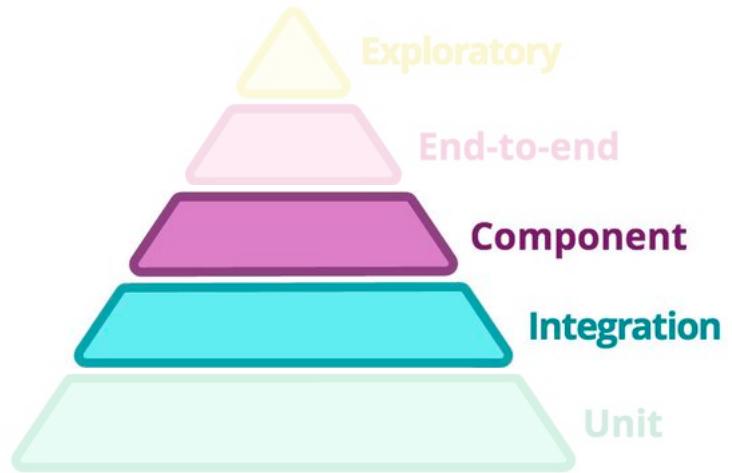
```
it('matches snapshot', () => {
  expect(renderer.create(
    <BirthdayForm
      selectedMonth={'March'}
      selectDay={'12'}
      zodiacSign={'Pisces'} />
  )).toMatchSnapshot();
});
```

```
it('calculates zodiac sign correctly', () => {
  const state = {
    selectedMonth: 'March',
    selectedDay: '12'
  };

  const props = mapStateToProps(state);

  expect(props.zodiacSign).toEqual('Pisces');
});
```





Snapshot Testing (The Good, The Bad)

An alternative
Integration

What is snapshot testing?

The Good

```
it('renders correctly', () => {
  const tree = renderer.create(
    <Link page="http://www.instagram.com">Instagram</Link>
  ).toJSON();

  expect(tree).toMatchSnapshot();
});
```

```
it('renders correctly', () => {
  const tree = renderer.create(
    <Link page="http://www.facebook.com">Facebook</Link>
  ).toJSON();

  expect(tree).toMatchSnapshot();
});
```

2. node

FAIL __tests__/Link.react-test.js

- **renders correctly**

```
expect(value).toMatchSnapshot()

Received value does not match stored snapshot 1.

- Snapshot
+ Received

<a
  className="normal"
- href="http://www.facebook.com"
+ href="http://www.instagram.com"
  onMouseEnter={[Function]}
  onMouseLeave={[Function]}>
- Facebook
+ Instagram
</a>

at Object.<anonymous> (__tests__/Link.react-test.js:14:16)
```

✗ **renders correctly** (10ms)

Snapshot Summary

› **1 snapshot test** failed in 1 test suite. Inspect your code changes or press `u` to update

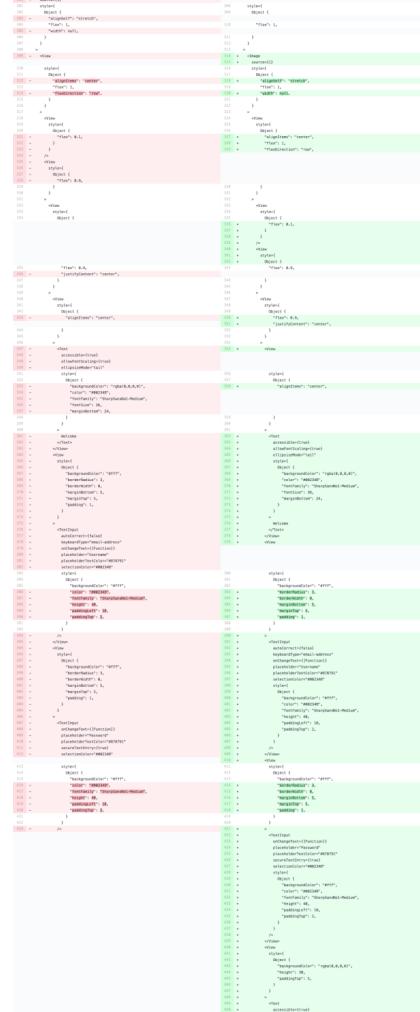
→ ~ jest --updateSnapshot



The Bad

```
it('Cowboy Horoscope homepage should render correctly', () => {
  expect(renderer.create(
    <App {...props} />
  )).toMatchSnapshot();
});
```

```
it('Cowboy Horoscope homepage should render correctly', () => {
  expect(renderer.create(
    <div>
      <App {...props} />
    </div>
  )).toMatchSnapshot();
});
```





LOOKS GOOD TO ME

What is snapshot testing
An alternative
Integration



Percy

Continuous visual reviews for web apps

Before



Milq n Wax / screenshot-test-app

App: default



GET YOUR COWBOY HOROSCOPE NOW

Enter the month and day of your birthday

MONTH

Select...

DAY

Select...

REVEAL YOUR COWBOY HOROSCOPE

Made with ❤ in NY by Carly Litchfield

After

Select width ▾



Approve

320px

1280px



GET YOUR COWBOY HOROSCOPE NOW

Enter the month and day of your birthday

MONTH

Select...

DAY

Select...

REVEAL YOUR COWBOY HOROSCOPE

Made with ❤ in NY by Carly Litchfield

Before



Milq n Wax / screenshot-test-app

App: default



COWBOY HOROSCOPE

UNLOCK THE WISDOM OF THE STARS, FARMERS, AND THE GREAT AMERICAN WEST.

GET YOUR COWBOY HOROSCOPE NOW

Enter the month and day of your birthday

MONTH

Select...

DAY

Select...

REVEAL YOUR COWBOY HOROSCOPE



Made with ❤ in NY by Carly Litchfield

After

Select width ▾



320px

1280px



COWBOY HOROSCOPE

UNLOCK THE WISDOM OF THE STARS, FARMERS, AND THE GREAT AMERICAN WEST.

GET YOUR COWBOY HOROSCOPE NOW

Enter the month and day of your birthday

MONTH

Select...

DAY

Select...

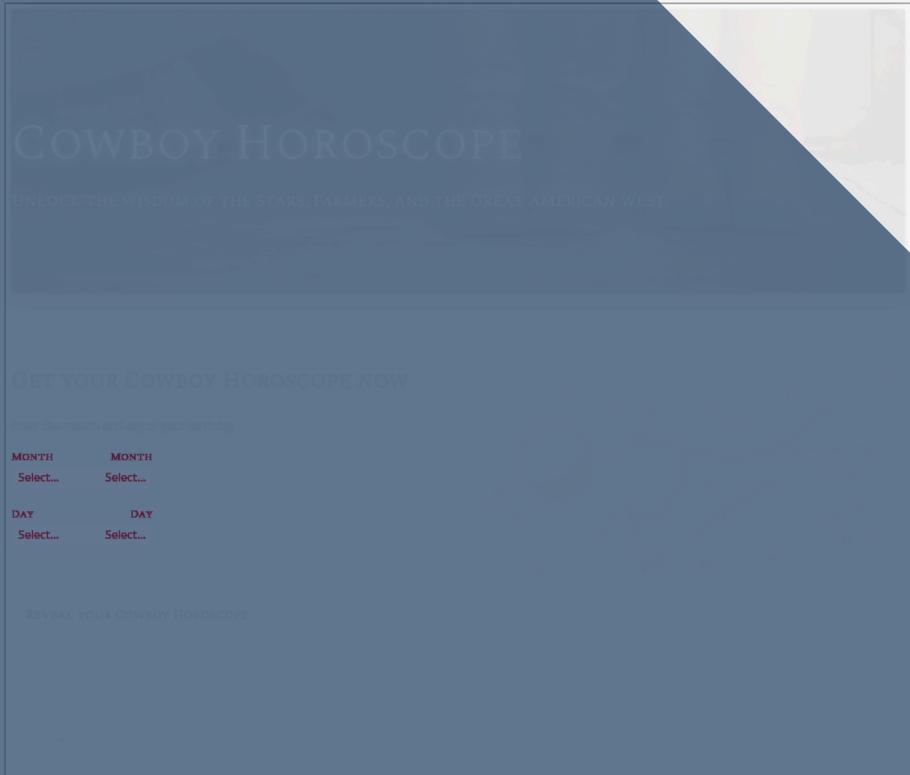
REVEAL YOUR COWBOY HOROSCOPE



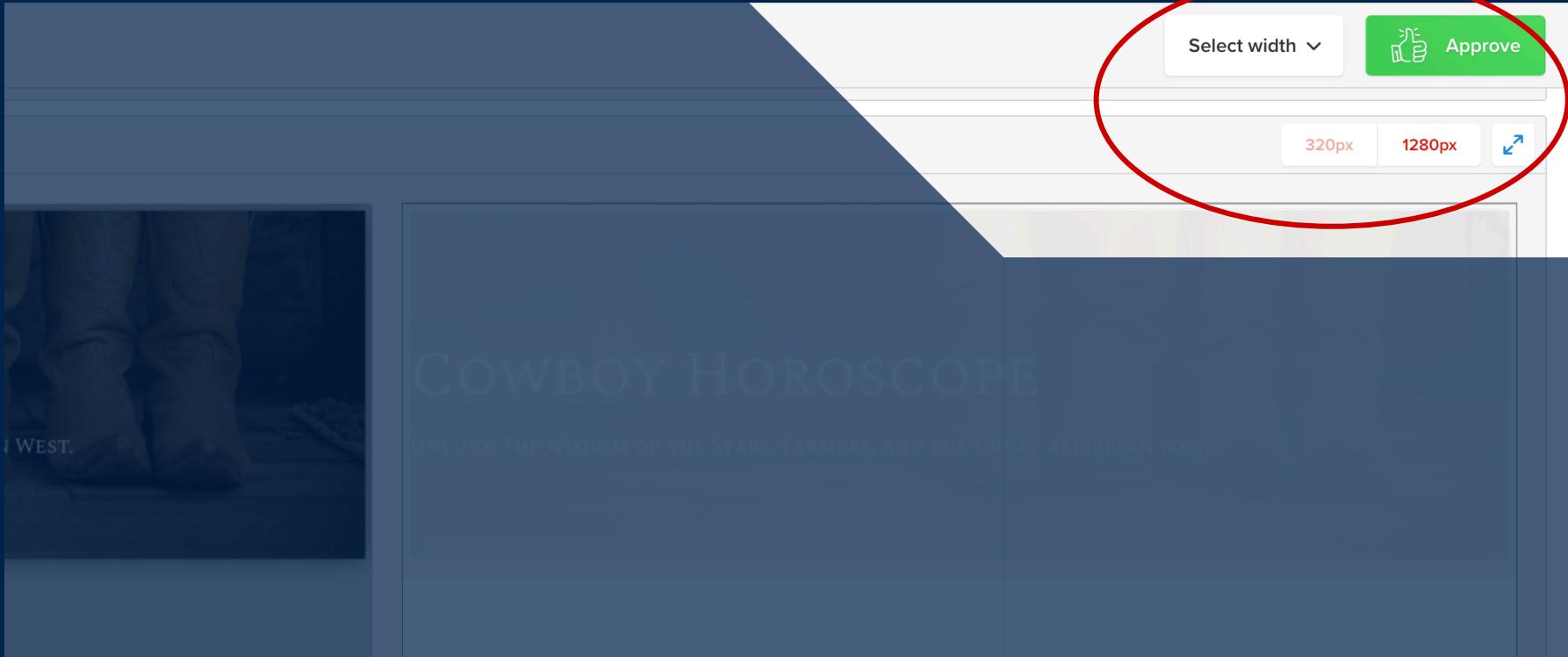
App: default

320px

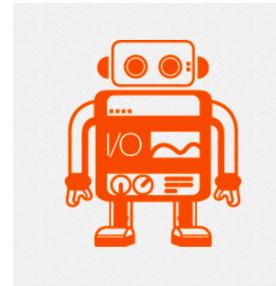
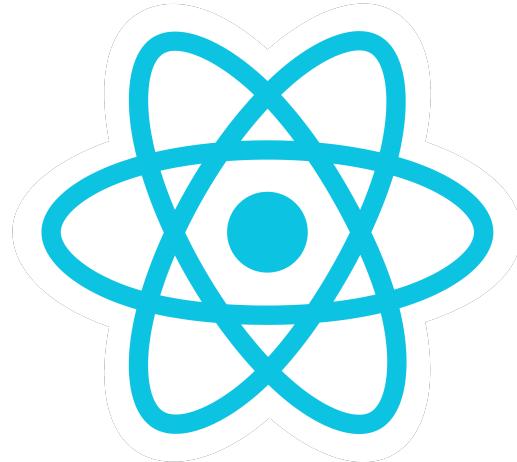
1280px



Test all your favorite screen sizes







**What is snapshot testing
An alternative
Integration**

Easy integration with React Storybook

<https://percy.io/docs/clients/javascript/react-storybook>
<https://github.com/cjlitch/screenshot-test-app>



REACT STORYBOOK



Next Story

Button

default view

link button

some emojis as the text

custom styles

REACT STORYBOOK



Filter

Button

default view

link button

some emojis as the text

custom styles

```
storiesOf('BirthdayForm', module)
  .add('default', () => <BirthdayForm />);
```

Your stories are your Percy snapshot tests

```
storiesOf('BirthdayForm', module)
  .add('default', () => <BirthdayForm />);
```

React Integration

<https://percy.io/docs/clients/javascript/react>

<https://github.com/cjlitch/screenshot-test-app>

```
percySnapshot('BirthdayForm', () => {
  return <BirthdayForm />
});
```

```
→ ~ npm run percy
```



Some checks were not successful

1 errored and 1 successful checks

[Hide all checks](#)

  [percy/booking-web-application](#) — 12 visual diffs need review

[Required](#) [Details](#)

  [continuous-integration/travis-ci/pr](#) — The Travis CI build passed

[Required](#) [Details](#)

Merging is blocked

Merging can be performed automatically with one approved review.

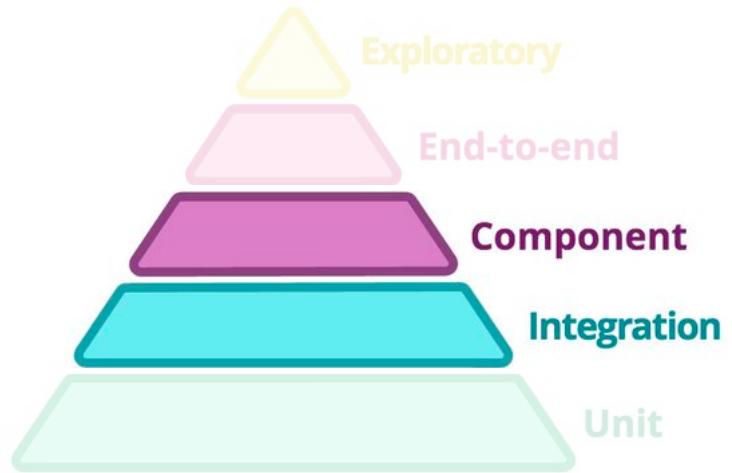
[Squash and merge](#)

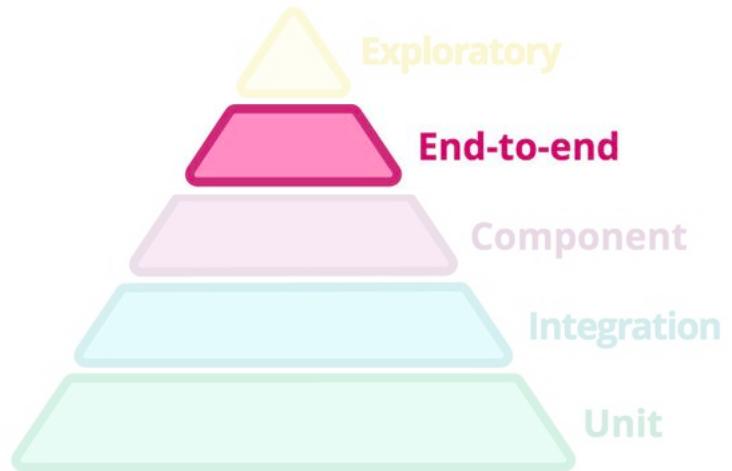
You can also [open this in GitHub Desktop](#) or view command line instructions.

What's the catch?

1. Doesn't catch everything

1. Doesn't catch everything
2. \$\$\$







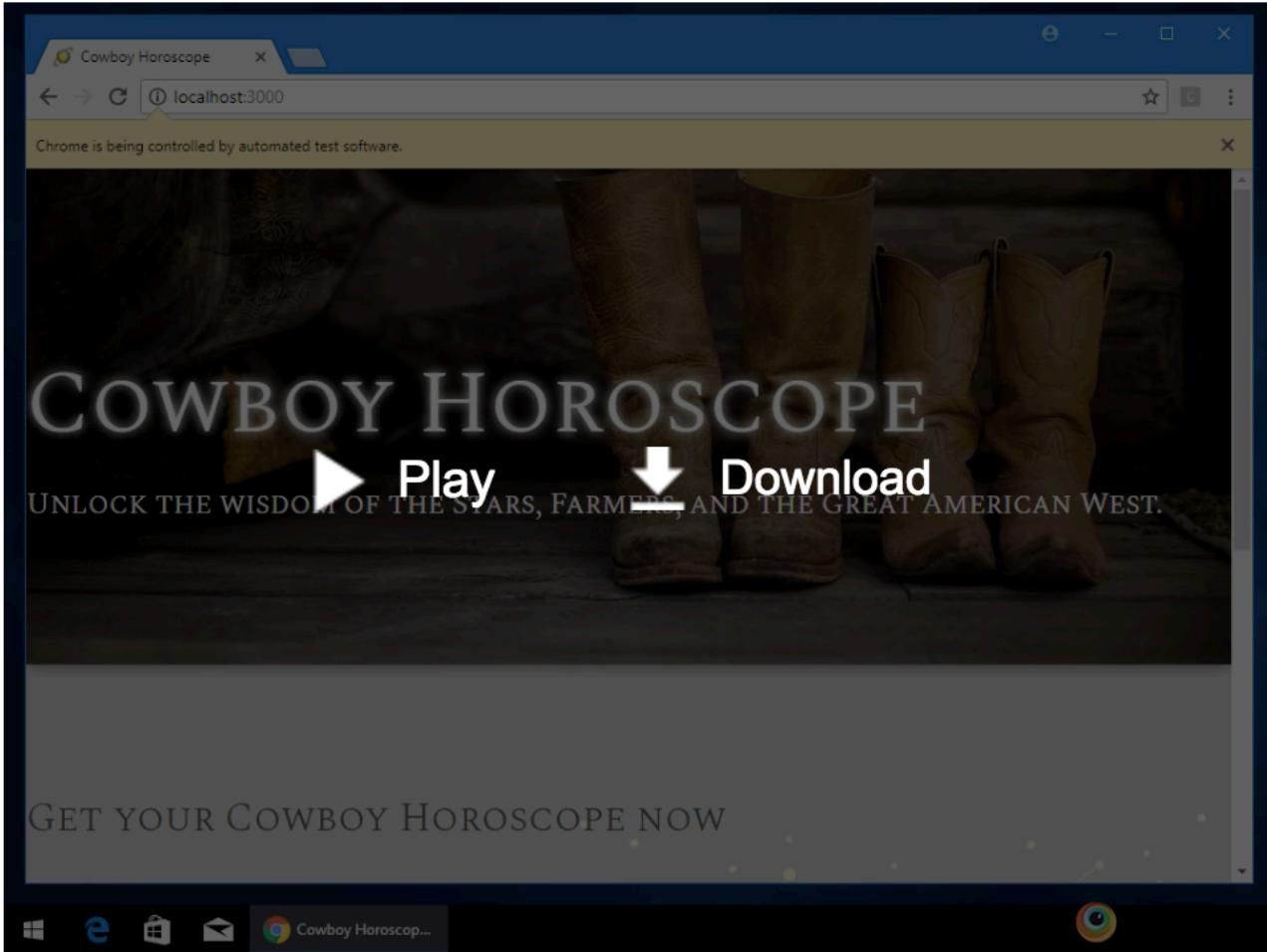
Last Line of Defense

WebdriverIO

```
it('should have the right title', () => {
  browser.url('http://localhost:3000');
  var title = browser.getTitle();
  assert.equal(title, 'Cowboy Horoscope');
});
```

Browserstack Automate





Capabilities Generator

Configure capabilities

Select browser / device

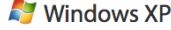
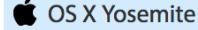
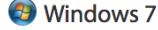
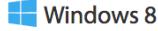
OS



Mobile



Desktop



Node JS

```
1 var capabilities = {  
2   'os_version' : '10.3',  
3   'device' : 'iPhone 7',  
4   'real_mobile' : 'true'  
5   'browserstack.selenium' :  
6   'browserstack.user' :  
7   'browserstack.key' : '  
8 }  
9
```

8
9
10

Resolution



1024x768



1024x768

1280x960

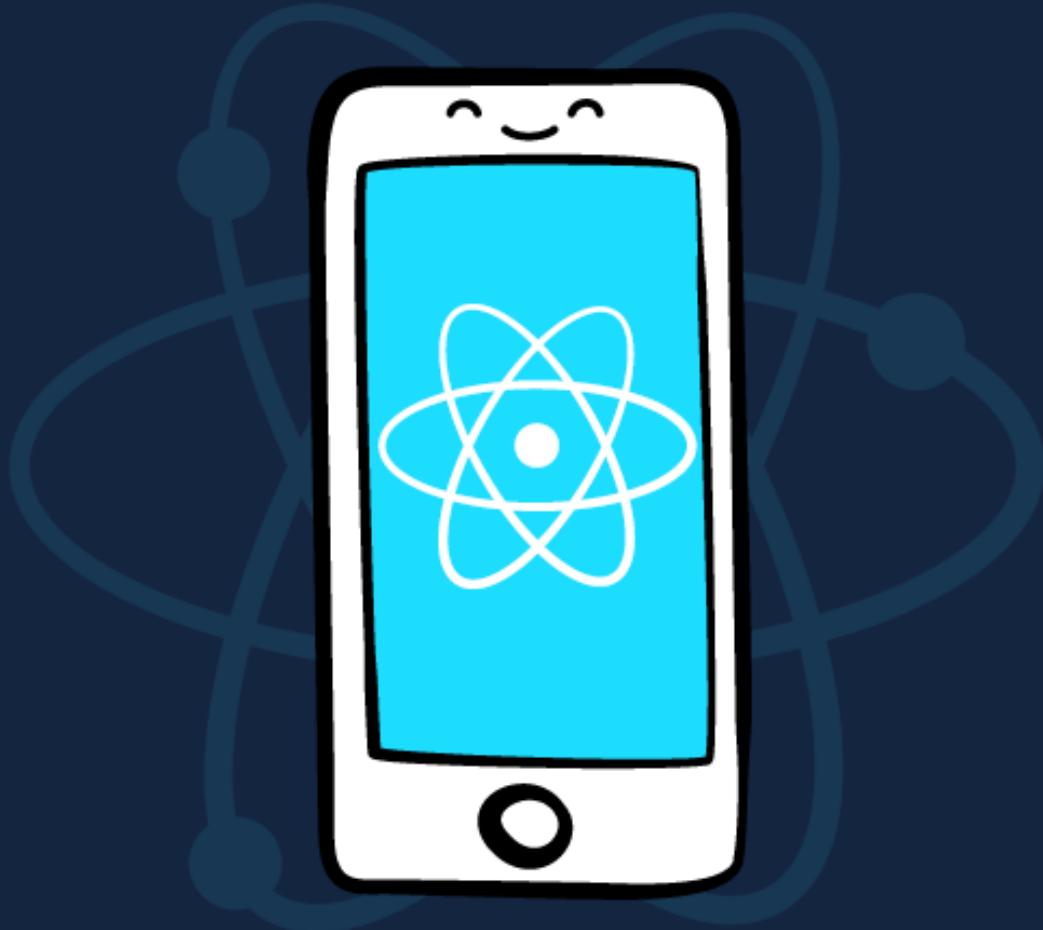
1280x1024

1600x1200

1920x1080

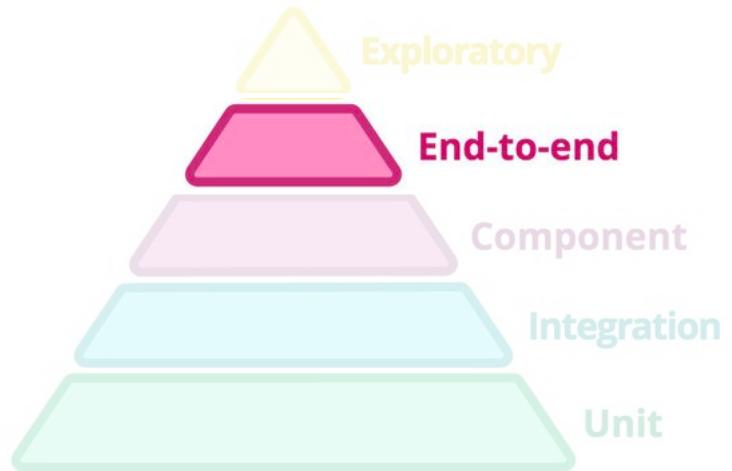
For 'Login Form'

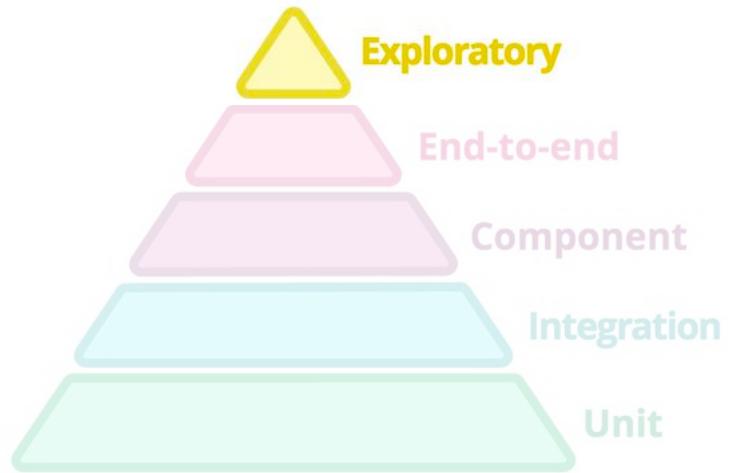
```
services: ['browserstack'],
```



Mock all the things

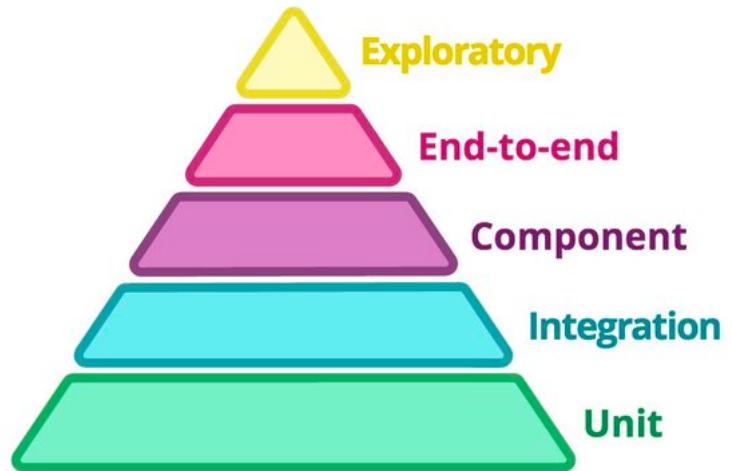
```
router.get('/horoscope', (req, res) => {
  res.json({ message: 'Fake horoscope' });
});
```





Browserstack Live





Why Test

How to Test

Conclusion

BABEL

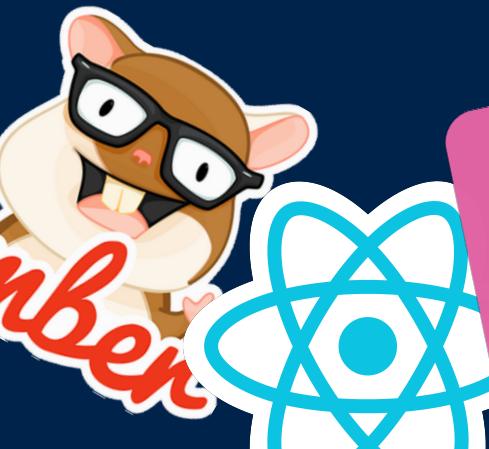


PREACT

METEOR



Front end Renaissance



Responsibility to Maintainability

Relevant Links

Cowboy Horoscope Code (and these slides): <https://github.com/cjitch/screenshot-test-app>

Cowboy Horoscope: <http://www.cowboyhoroscope.com/>

Jest: <https://facebook.github.io/jest/>

Enzyme: <https://github.com/airbnb/enzyme>

Jest Snapshot Docs: <https://facebook.github.io/jest/docs/en/snapshot-testing.html>

Percy: <https://percy.io/>

Percy with React Storybook: <https://percy.io/docs/clients/javascript/react-storybook>

Percy with React: <https://percy.io/docs/clients/javascript/react>

Browserstack: <https://www.browserstack.com>

WebdriverIO: <http://webdriver.io/>

Travis for Open Source: <https://travis-ci.org/>

Feedback

<http://bit.ly/testing-ftw>



zocdoc.com/careers

Thank you!

Carly Litchfield
Zocdoc