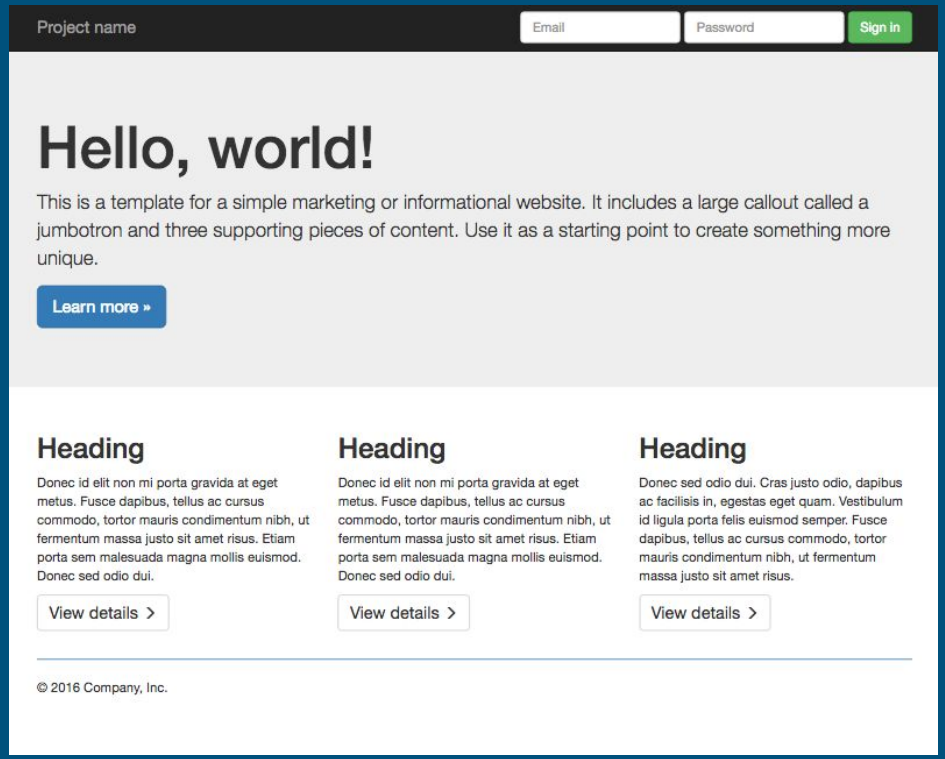
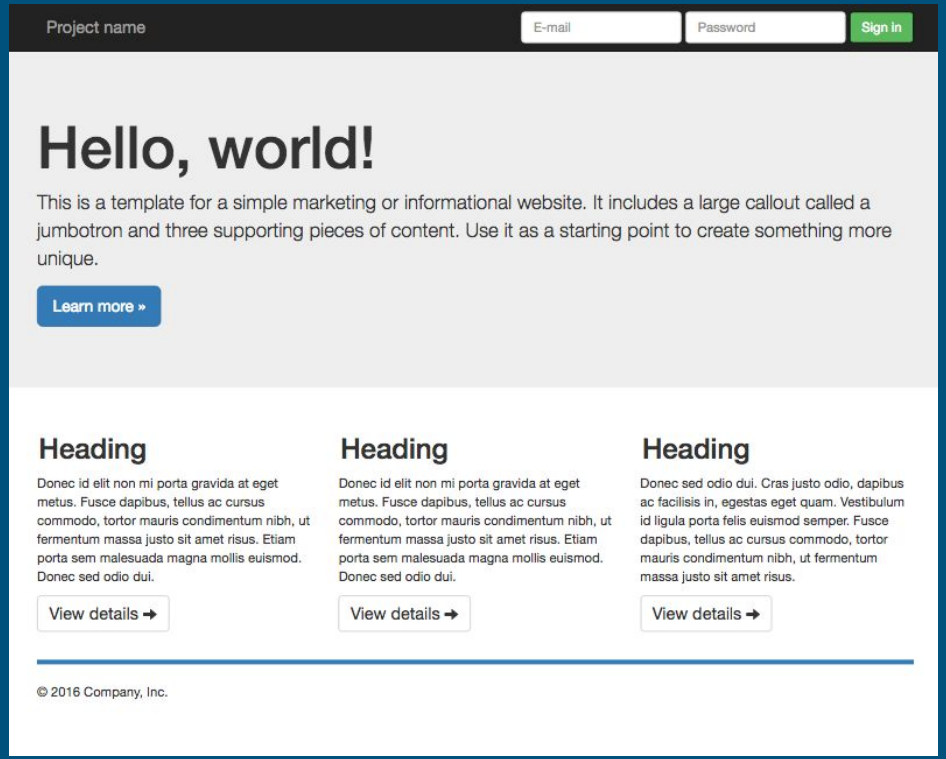
Visual Regression Testing

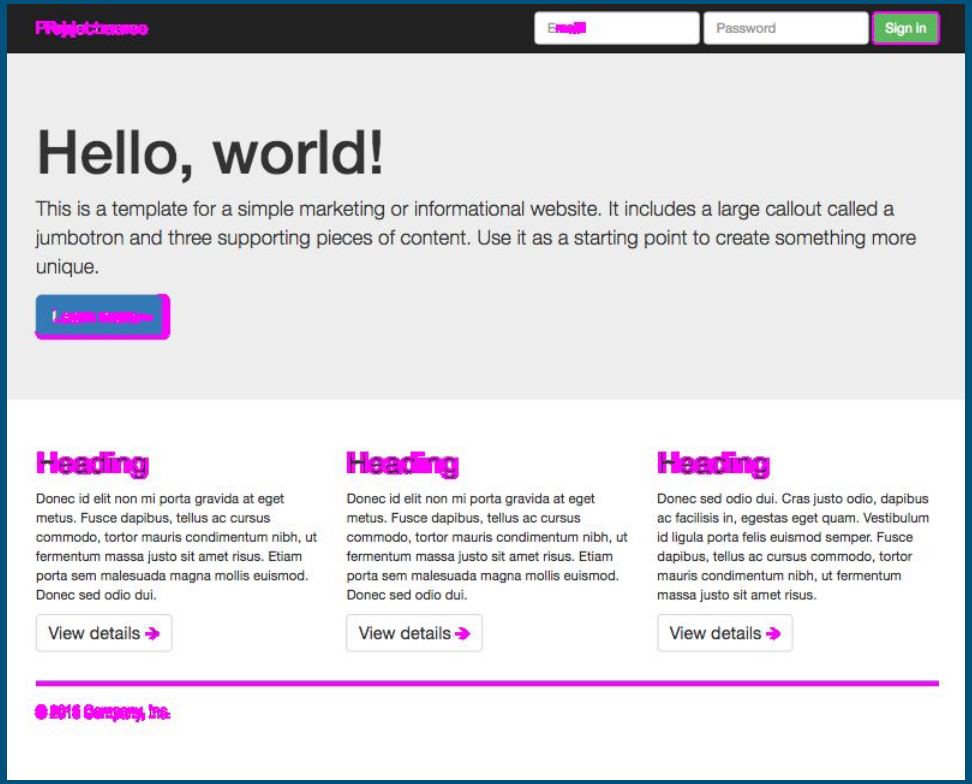
Adela Istrate

Spot the difference

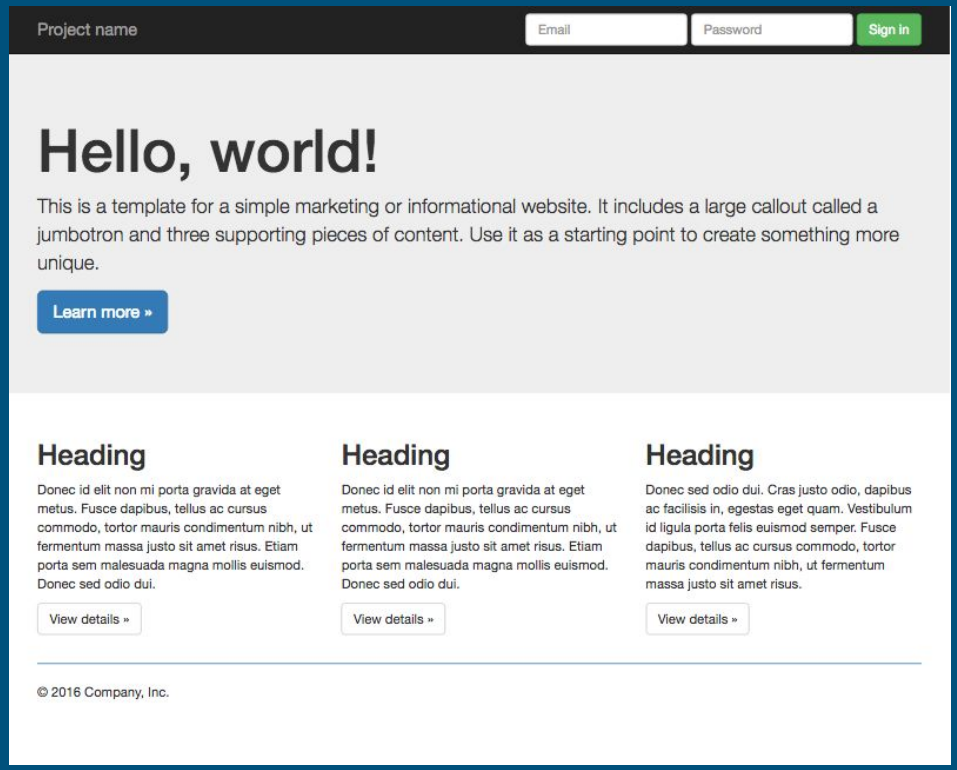


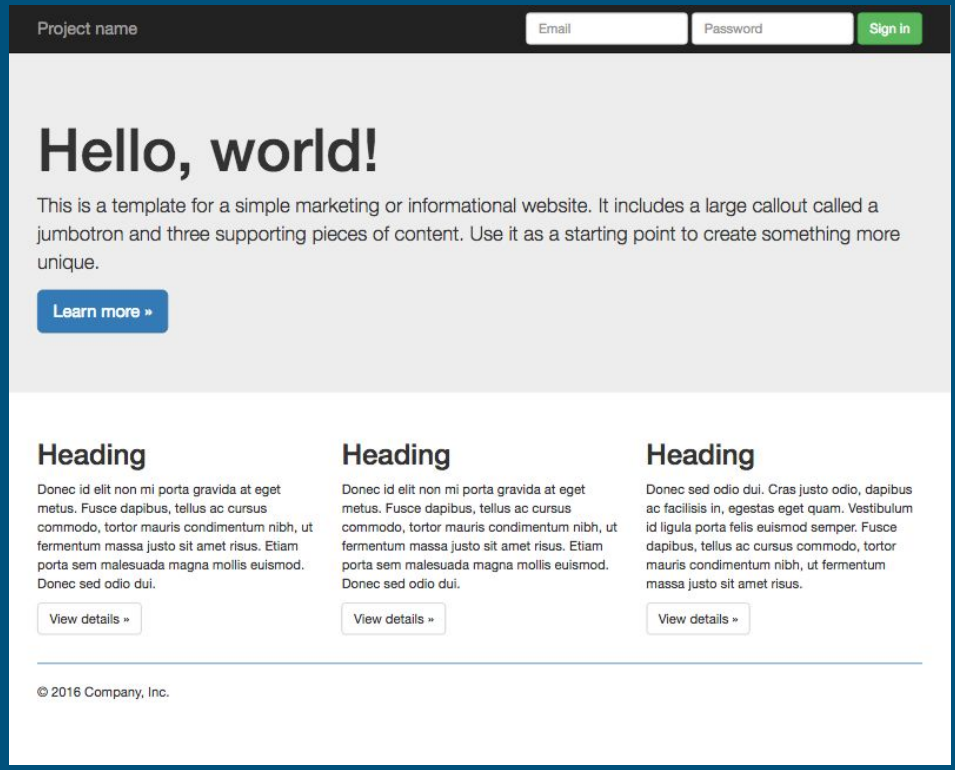


Spot the difference

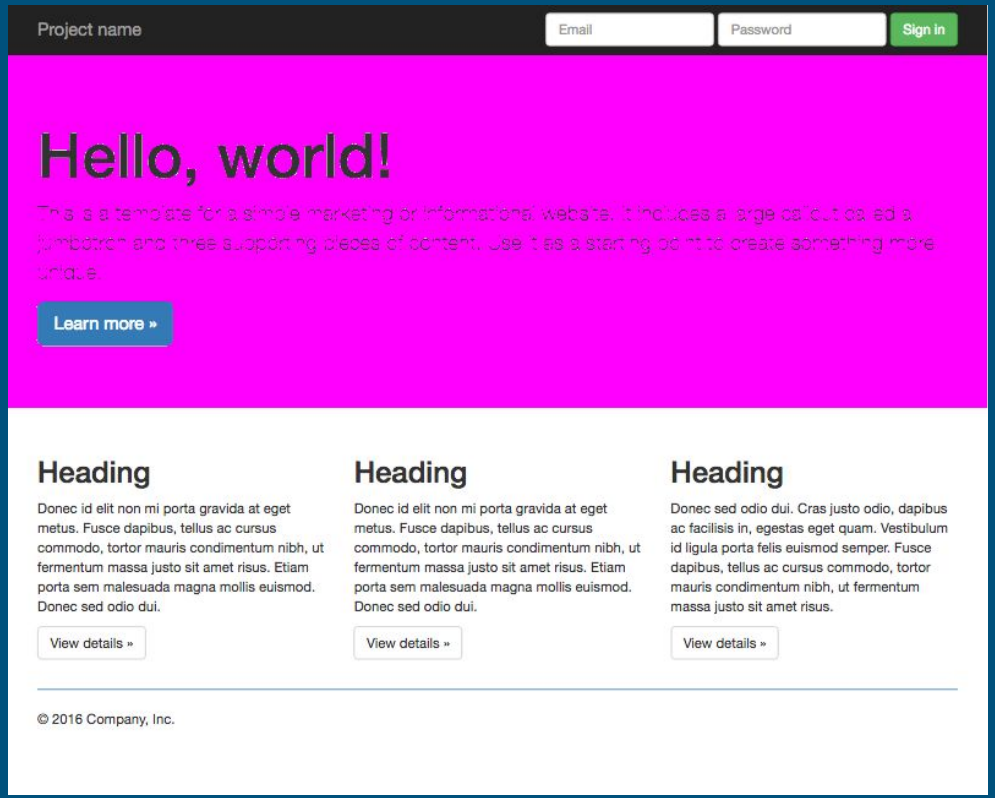


Spot the difference





Spot the difference



Visual regression testing

Run the app

Take screenshot

Have a screenshot of the previous version

Compare and generate diffs

Atomic testing

think about what you can really test?

set up a playground

test those components

Mugshot

library for visual regression testing

Why Mugshot?

Other tools:

- PhantomJS -> old webkit
- PhantomCSS -> false positives

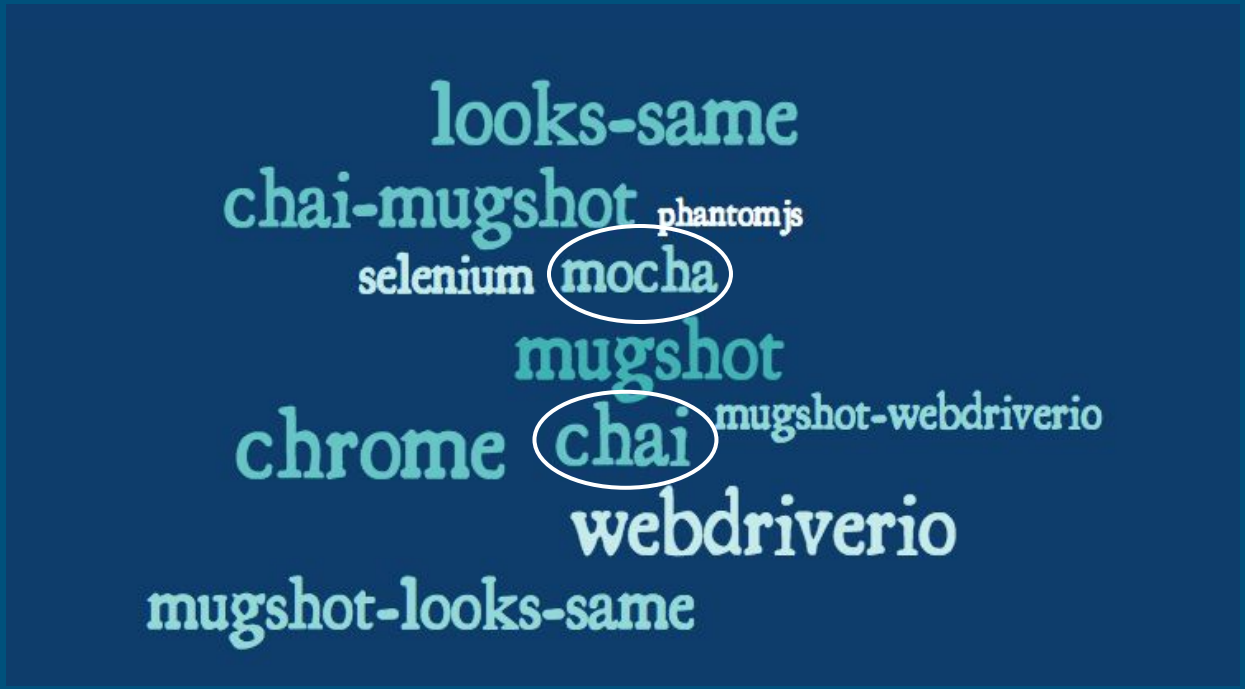
Independent of any browser, test-runner, diffing tool …

- achieves this through its interfaces

Mugshot - how it works?

baseline	screenshot	diff image
pic.png	pic.new.png	pic.diff.png

Mugshot - what do you need?



Mocha

JavaScript test framework

asynchronous testing is simple and fun

natural description of the tests

Mocha

```
var expect = require('chai').expect;¬
¬
describe('Dummy tests', function() {¬
····it('should include number 2', function() {¬
·····expect([1,2,3]).to.include(2);¬
····});¬
····it('should include the string foo', function() {¬
·····expect('foobar').to.contain('foo');¬
····});¬
····it('should be true', function() {¬
·····expect(false).to.be.true;¬
····});¬
});¬
```

Chai

Assertion Library

can be paired with any javascript testing framework

several interfaces

Chai

Should

```
chai.should();

foo.should.be.a('string');
foo.should.equal('bar');
foo.should.have.length(3);
tea.should.have.property('flavors')
  .with.length(3);
```

Expect

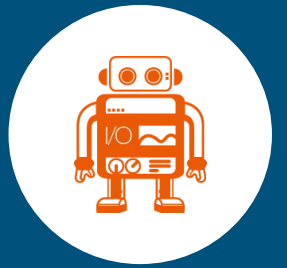
```
var expect = chai.expect;

expect(foo).to.be.a('string');
expect(foo).to.equal('bar');
expect(foo).to.have.length(3);
expect(tea).to.have.property('flavors')
  .with.length(3);
```

Assert

```
var assert = chai.assert;

assert.typeOf(foo, 'string');
assert.equal(foo, 'bar');
assert.lengthOf(foo, 3)
assert.property(tea, 'flavors');
assert.lengthOf(tea.flavors, 3);
```

Mugshot - what do you need?

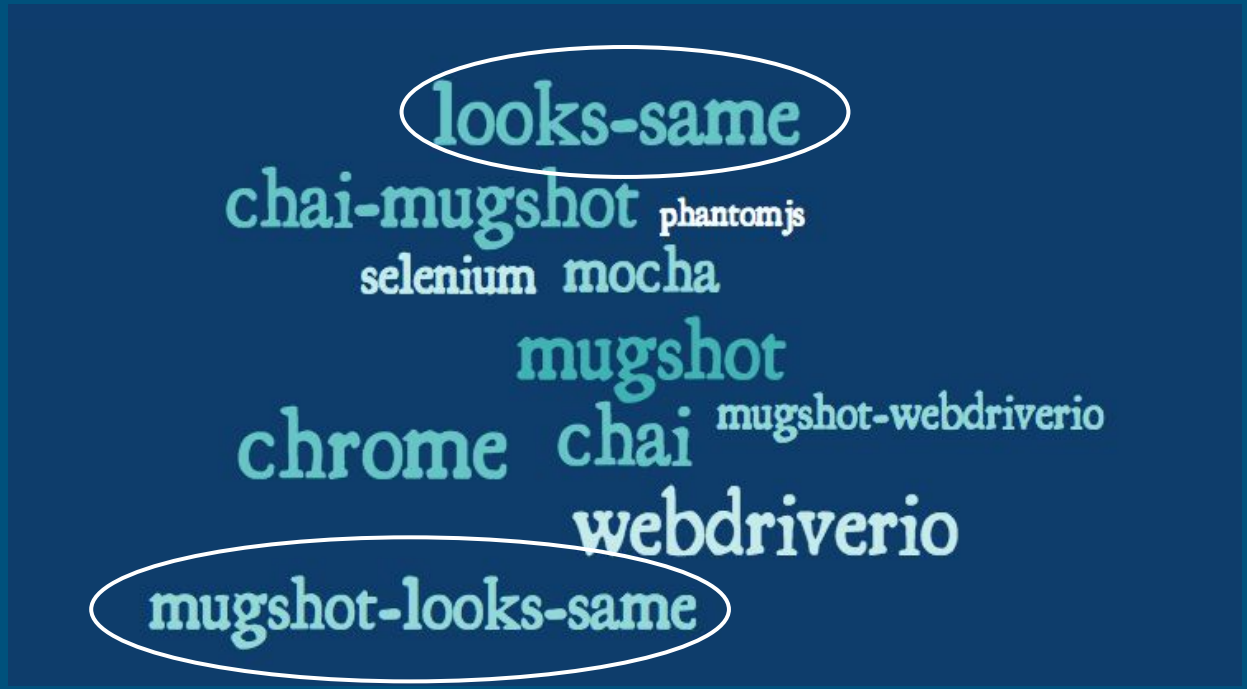
Mugshot-WebDriverIO



Mugshot adapter for WebDriverIO

```
before(function(done) {
  var options = {
    desiredCapabilities: {
      browserName: 'chrome'
    }
  };

  // 1. Prepare the browser instance.
  webdriverioInstance = webdriverio.remote(options).init()
    .url('https://hootsuite.com')
    .then(function() {
      // 2. Create an adapter over it.
      var browser = new WebdriverIOAdapter(webdriverioInstance);

      // 3. Hand it over to Mugshot.
      mugshot = new Mugshot(browser);

      done();
    });
});
```

Mugshot - what do you need?



Mugshot-Looks-Same

Mugshot adapter for Looks Same

-> a library for comparing PNG-images, taking into account human color perception

Mugshot example

```
30    it('should something', function(done) {
31      var captureItem = {
32        name: 'hootsuite-home-page',
33        selector: '#header'
34      };
35
36      // 4. Do the testing.
37      mugshot.test(captureItem, function(error, result) {
38        expect(error).to.be.null;
39        expect(result.isEqual).to.be.true;
40
41        done();
42      });
43    });
44
```

Mugshot example

```
adela-istrate:v2 adela-istrate$ mocha test.js --timeout 6000


  Suite name
    1) should be the same


  0 passing (924ms)
  1 failing

  1) Suite name should be the same:
     Uncaught AssertionError: expected false to be true
      at test.js:46:35
      at /Users/adela-istrate/ubervu/article/node_modules/mugshot/lib/mugshot.js:181:7
      at /Users/adela-istrate/ubervu/article/node_modules/graceful-fs/graceful-fs.js:43:10
      at FSReqWrap.oncomplete (fs.js:82:15)



adela-istrate:v2 adela-istrate$
```

Chai-Mugshot

plugin for pretty output

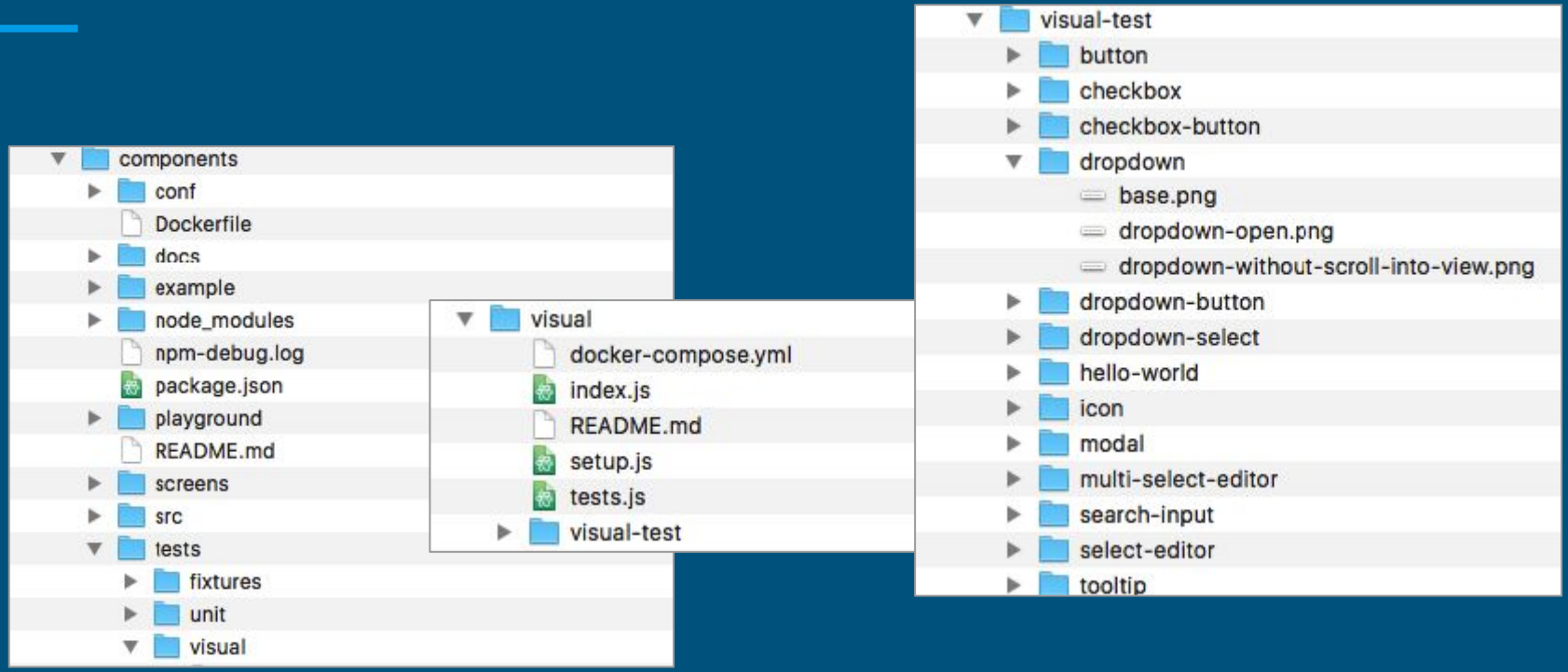
```
35
36        chai.use(chaiMugshot(mugshot));
37
```

```
47
48      // 4. Do the testing.
49      return expect(captureItem).to.be.identical;
```

Chai-Mugshot

```
adela-istrate:v3 adela-istrate$ mocha test.js


  Suite name
    1) should be the same


  0 passing (920ms)
  1 failing

  1) Suite name should be the same:
     AssertionError: expected baseline and screenshot of 'base' ([class^="component-playground__preview"] > *) to be identical



adela-istrate:v3 adela-istrate$
```

???


components
conf
Dockerfile
docs
example
node_modules
npm-debug.log
package.json
playground
README.md
screens
src
tests
fixtures
unit
visual
visual
docker-compose.yml
index.js
README.md
setup.js
tests.js
visual-test
visual-test
button
checkbox
checkbox-button
dropdown
base.png
dropdown-open.png
dropdown-without-scroll-into-view.png
dropdown-button
dropdown-select
hello-world
icon
modal
multi-select-editor
search-input
select-editor
tooltip

Mocha-Mugshot-Reporter

dedicated interface for visual regression tests

filter tests by passed/failed

displays the diffs in a `GitHub style`

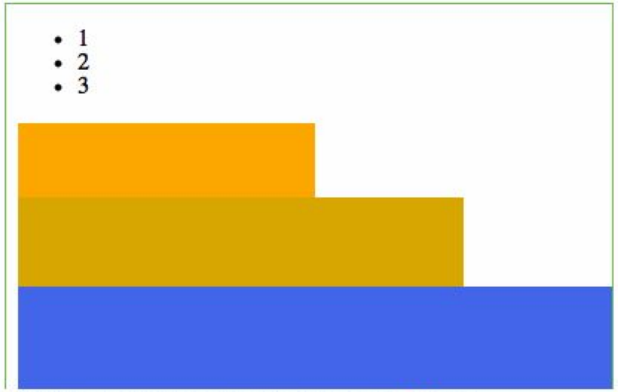
Mocha-Mugshot-Reporter

All 4 | Passes 3 | Failures 1 | Duration 470 ms

Generate a dummy report for testing

First suite

✔ should be ok : *passed* in 191 ms



✔ should pass and generate diffs : *passed* in 177 ms

Second suite

✖ should fail with 'No browser provided' error : *failed* in 0 ms

✔ should be ok : *passed* in 102 ms

Demo

Thank you