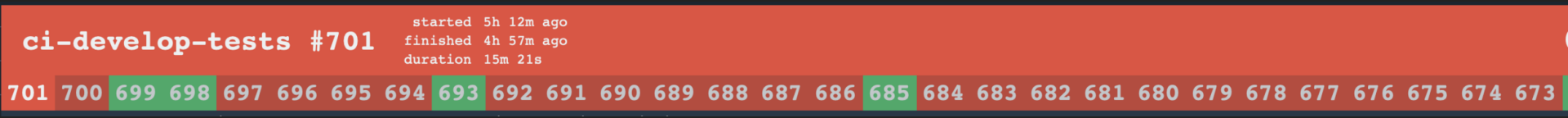
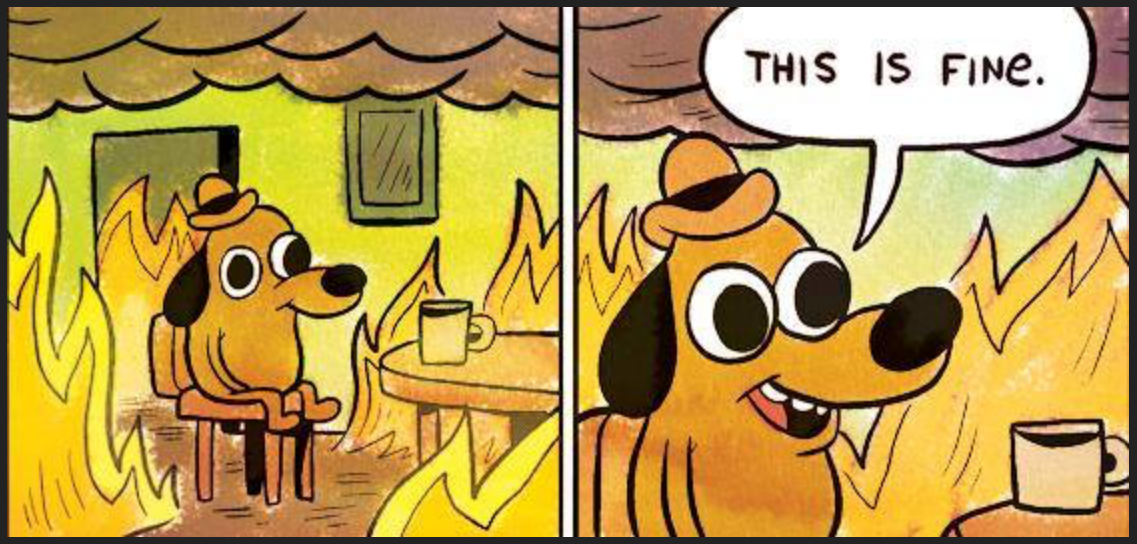
Resilient UI System Testing with Puppeteer

Mike Eastes

Senior Software Engineer, Code42

Seeing Red


ci-develop-tests #701
started 5h 12m ago
finished 4h 57m ago
duration 15m 21s
701 700 699 698 697 696 695 694 693 692 691 690 689 688 687 686 685 684 683 682 681 680 679 678 677 676 675 674 673


THIS IS FINE.

Issues

- Builds were slow
- Testing required a full environment
- Complex system interactions led to inconsistency

If tests are so flaky,
why bother at all?

Automated Tests can...

- Catch JavaScript errors that happen at runtime
- Catch edge cases in UI behavior
- Validate that data makes it from the API to being rendered
- Validate correctness of rare UI states (server errors)

Most important testing attribute

Credibility

D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/728 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/727 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/722 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/721 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/720 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/719 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/718 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/717 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/716 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/715 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/714 ...
D	devops	ci-develop-tests failed: forensic-search-develop/ci-develop-tests/713 ...

Know when to cut your losses

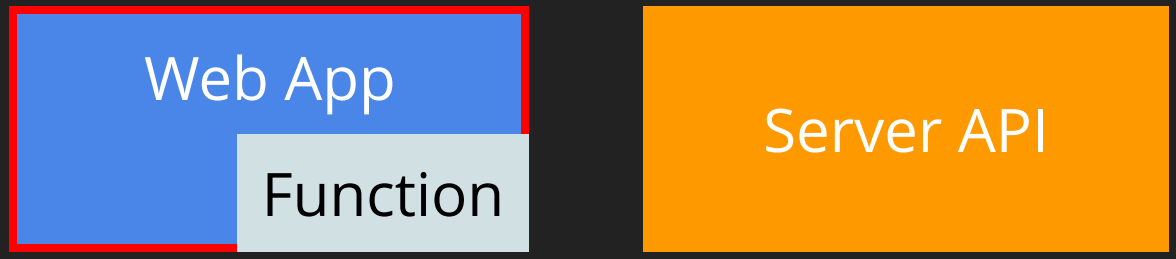
First: Terminology

Unit Test

“ Validates the correctness of a single unit of work (ie. function, interface).

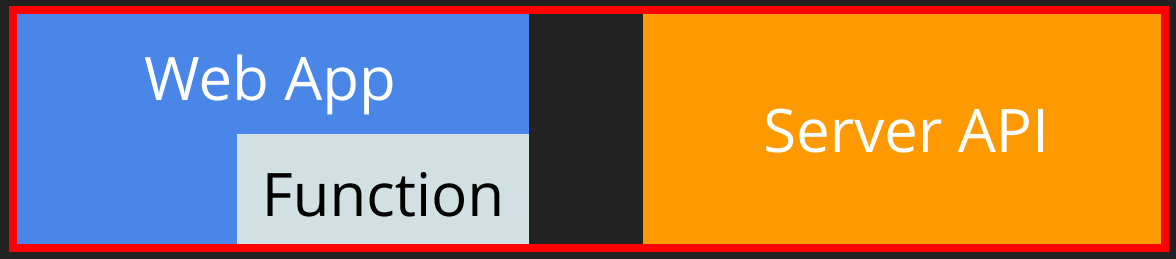
System Test

" Validates the correctness of a complete system including all modules (ie. visual components, state management, "glue code").



Acceptance Test

" Validates the correctness of complete service including all APIs and applications (ie. web UI, server API).



System vs. Acceptance

System tests

- Mock all data from server APIs
- Execute in a headless browser
- Validate correctness of the UI only

Acceptance tests

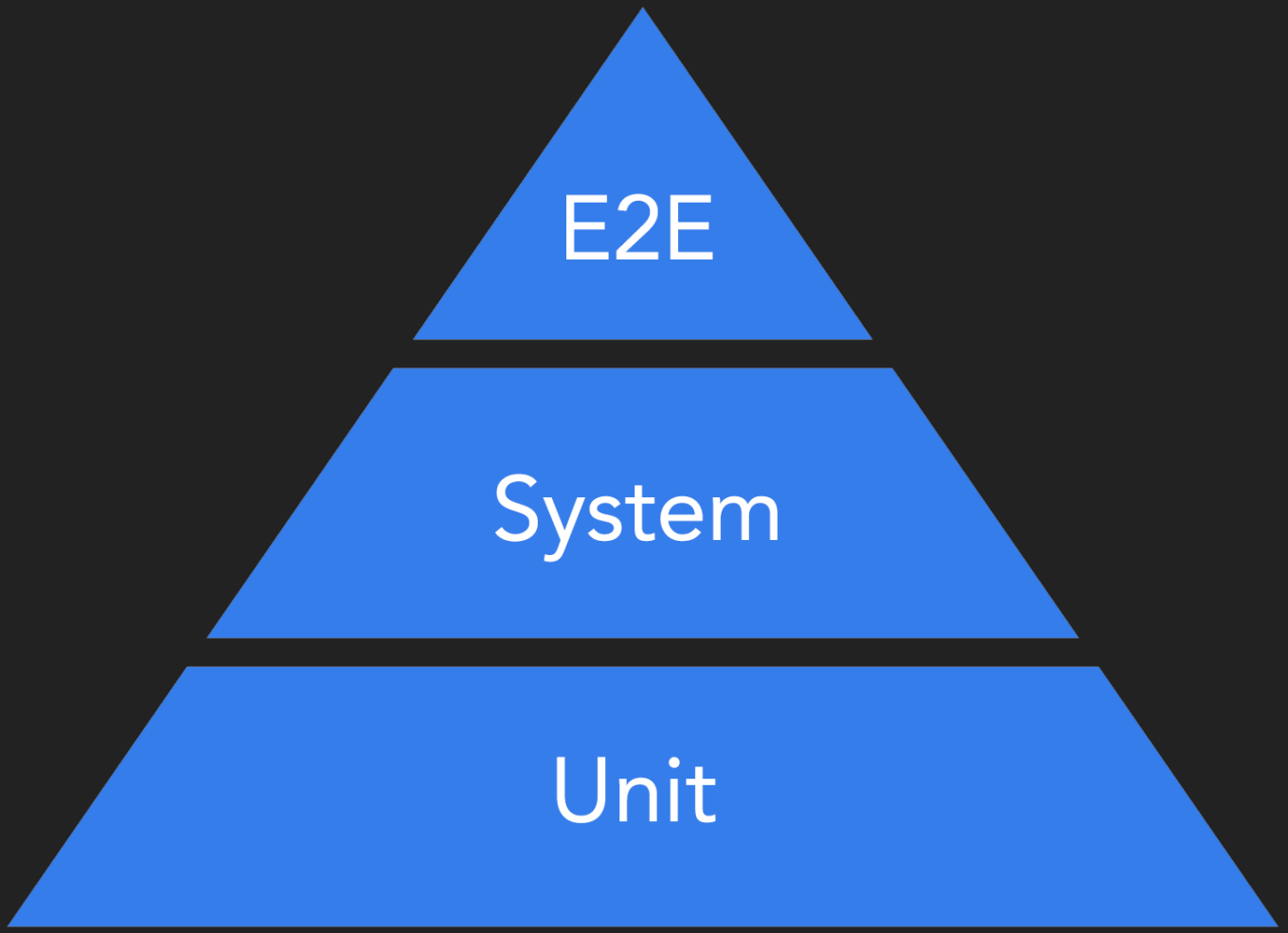
- Require real data
- Often execute in a full browser
- Validate correctness of entire system

One big system test advantage:

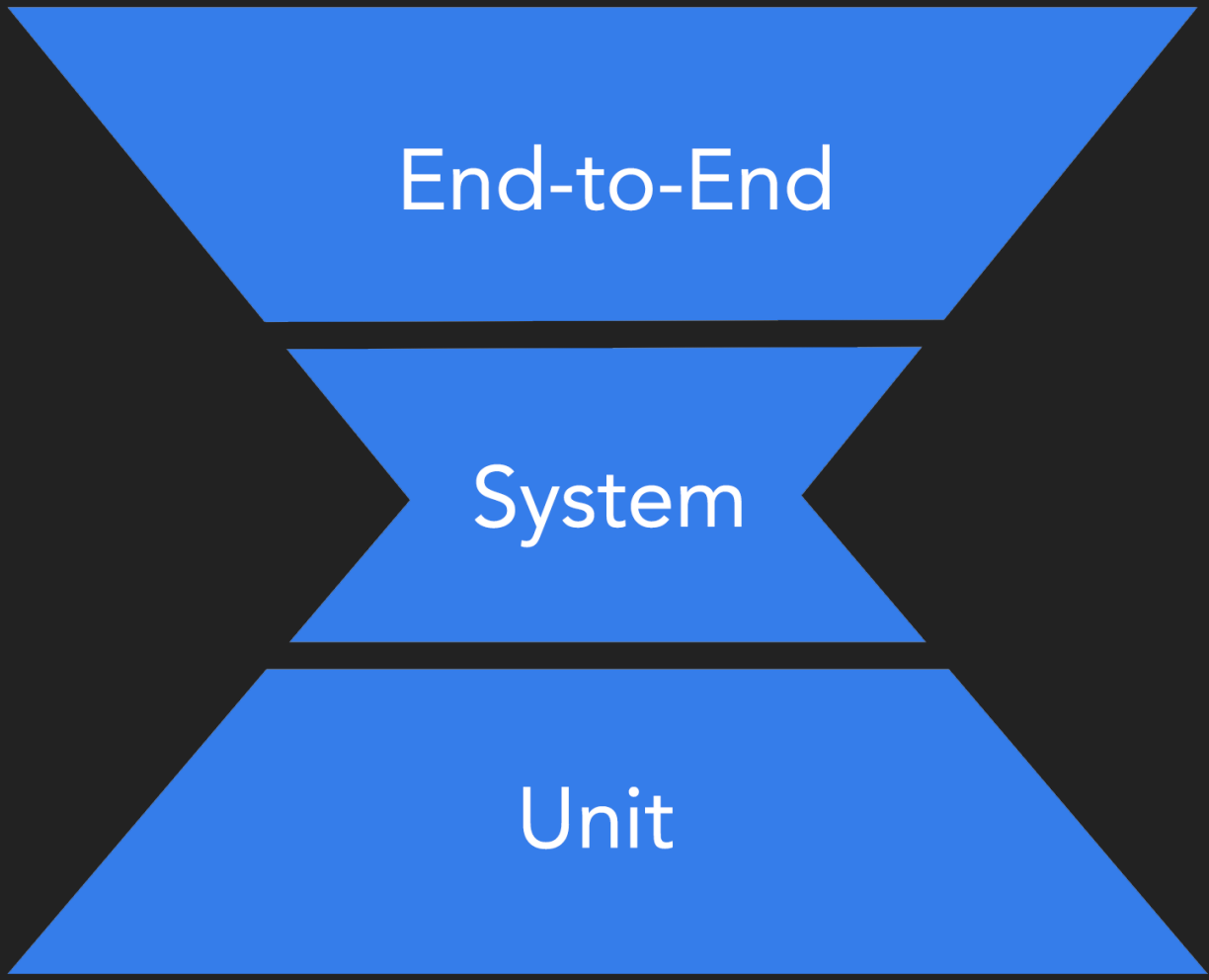
Shorter Feedback Cycles

How did we get here?

Test Pyramid



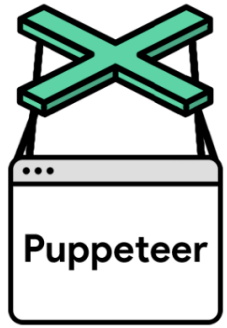
"Test Hourglass"



Puppeteer saves the day

Puppeteer is a Node library provided by Google that allows you to control a webpage inside of a (headless) instance of Chrome.

Puppeteer



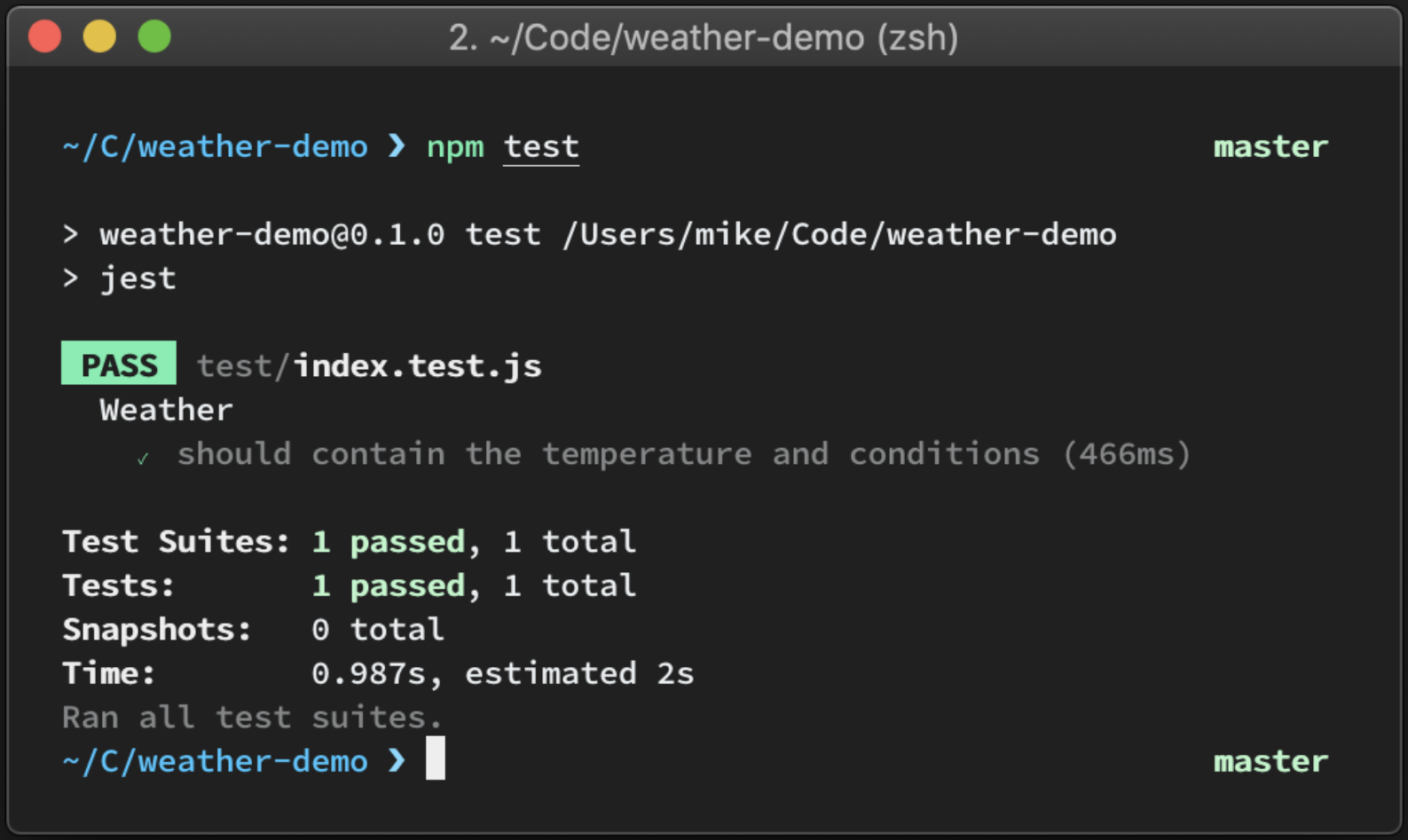
API | FAQ | Contributing | Troubleshooting

> Puppeteer is a Node library which provides a high-level API to control Chrome or Chromium over the DevTools Protocol. Puppeteer runs headless by default, but can be configured to run full (non-headless) Chrome or Chromium.

What can I do?

Most things that you can do manually in the browser can be done using Puppeteer! Here are a few examples to get you started:

- Generate screenshots and PDFs of pages.
- Crawl a SPA (Single-Page Application) and generate pre-rendered content (i.e. "SSR" (Server-Side Rendering)).
- Automate form submission, UI testing, keyboard input, etc.
- Create an up-to-date, automated testing environment. Run your tests directly in the latest version of Chrome


2. ~/Code/weather-demo (zsh)
~/C/weather-demo ❯ npm test
master
> weather-demo@0.1.0 test /Users/mike/Code/weather-demo
> jest
PASS test/index.test.js
Weather
✓ should contain the temperature and conditions (466ms)
Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.987s, estimated 2s
Ran all test suites.
~/C/weather-demo ❯
master

Example Test

```
describe('Google', () => {
  const Selector = {
    SearchField: 'input[type="text"]',
  };

  beforeEach(async () => {
    await page.goto('https://google.com');
  });

  it('should contain a search field', async () => {
    const searchInput = await page.$(Selector.SearchField);
    expect(searchInput).toBeTruthy();
  });
});
```

Example with Mock API

Current Minneapolis Weather

Temperature: 1.31

Conditions: Clear

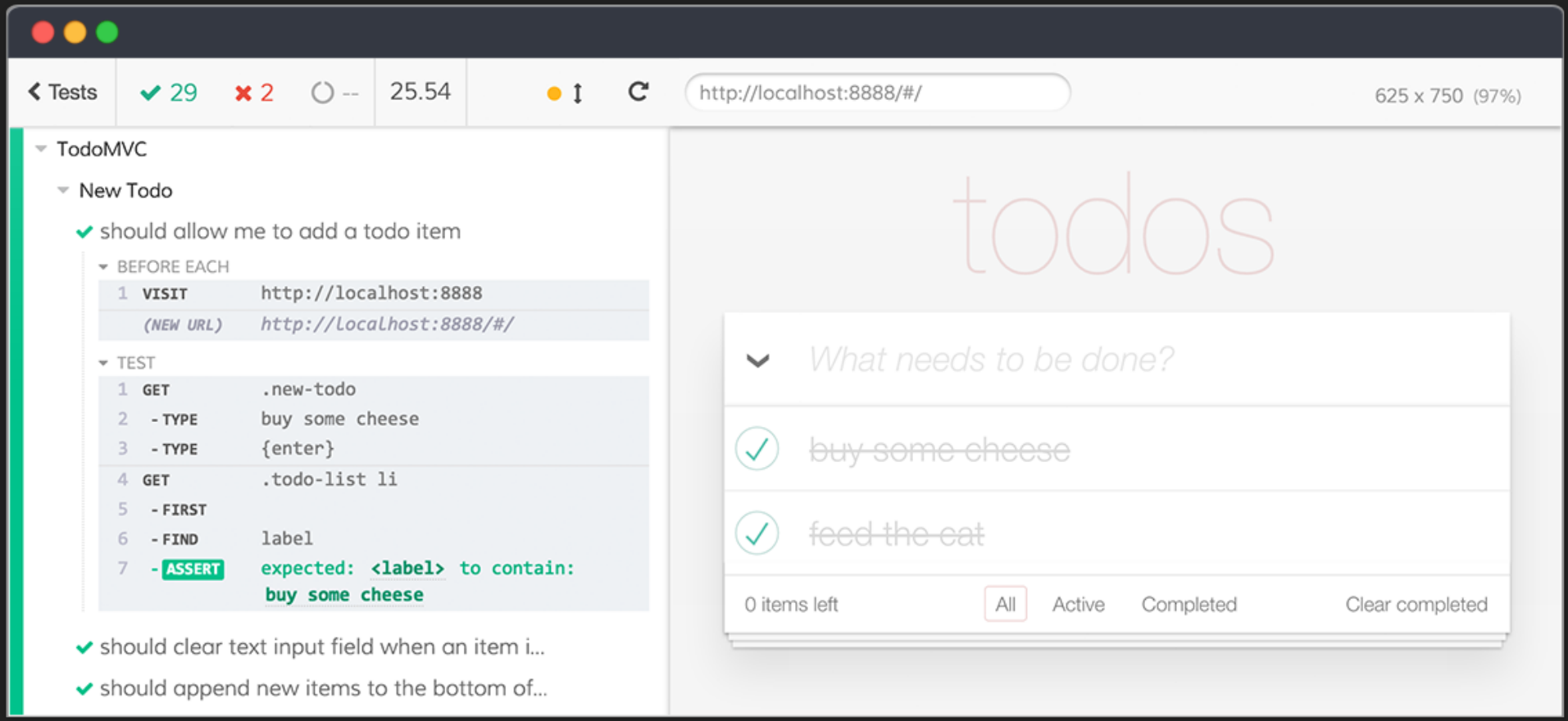
Powered by Dark Sky

```
describe('Weather', () => {
  const Selector = {
    Temperature: '[data-test="temperature"]',
    Conditions: '[data-test="conditions"]',
  };

  const url = 'http://localhost:3000';

  function mockResponse(data) {
    page.on('request', (interceptedRequest) => {
      if (interceptedRequest.url().endsWith('weather/get')) {
        interceptedRequest.respond({
          status: 200,
          contentType: 'application/json; charset=utf-8',
          headers: { 'Access-Control-Allow-Origin': '*' },
          body: JSON.stringify(data),
        });
      } else {
        interceptedRequest.continue();
      }
    });
  }

  beforeEach(() => {
    page.setRequestInterception(true);
  });

  it('should contain the temperature and conditions', async () => {
    mockResponse({ currently: { temperature: 42, summary: 'Sunny' } });
    await page.goto(url);

    await page.waitFor(Selector.Temperature);
    await page.waitFor(Selector.Conditions);

    const temp = await page.$(Selector.Temperature);
    expect(await temp.$eval('*', node => node.innerText)).toBe('42');

    const conditions = await page.$(Selector.Conditions);
    expect(await conditions.$eval('*', node => node.innerText).toBe('Sunny');
```

Other Puppeteer Features

- Generate images and PDFs from webpages
- Compatible with any JavaScript test runner
- Fully emulates a keyboard and mouse
- The ability to slow down tests for easier debugging
- Uses the real Chrome browser

All in One Solution: Cypress


Tests
29
2
--
25.54
http://localhost:8888/#/
625 x 750 (97%)
TodoMVC
New Todo
should allow me to add a todo item
BEFORE EACH
1 VISIT http://localhost:8888
(NEW URL) http://localhost:8888/#/
TEST
1 GET .new-todo
2 - TYPE buy some cheese
3 - TYPE {enter}
4 GET .todo-list li
5 - FIRST
6 - FIND label
7 - ASSERT expected: <label> to contain: buy some cheese
should clear text input field when an item i...
should append new items to the bottom of...
todos
What needs to be done?
buy some cheese
feed the cat
0 items left
All
Active
Completed
Clear completed

Example

```
describe('Website tests', () => {
  it('shows content when clicking the button', () => {
    cy.visit('https://www.mywebsite.com');

    cy.get('.button').click();

    cy.get('.content').should('exist');
  });
});
```

Includes many nice features

- Screenshots / videos
- Utility functions
- Test debugger

Downsides

- Less control
- More bugs
- Need to write tests in a different way (no variables!)

Would I recommend Cypress?

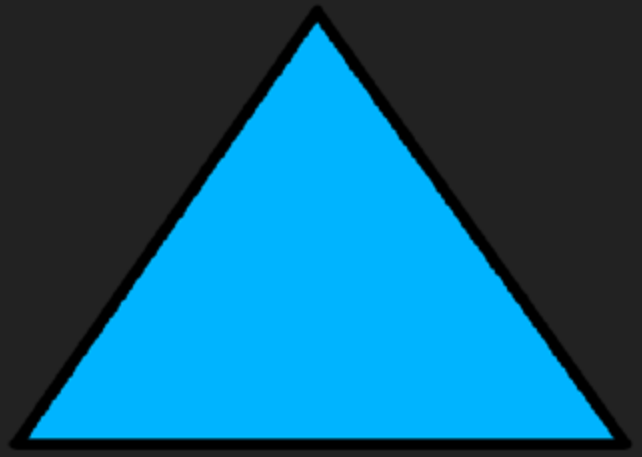
It depends...

Tips to Improve Reliability

Avoid hard-coded waits

Multiple assertions are okay!

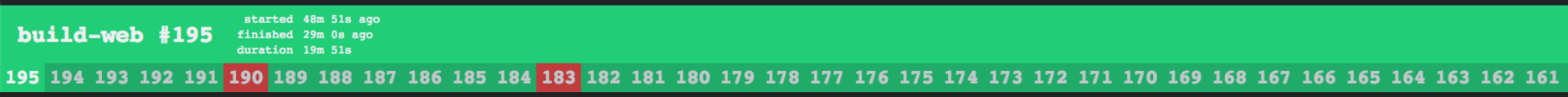
Consider your comprehensive test coverage



Results

Improvements

- Builds complete in less than half the time
- Tests run on every PR; only running the web app is required
- Reduced complexity and shorter feedback cycle resulted in less random failures


build-web #195
started 48m 51s ago
finished 29m 0s ago
duration 19m 51s
195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161

Where next?

Continuous Delivery

Following development practices that
allow you to release at any time.

Mike Eastes

mike.eastes@code42.com
https://github.com/meastes