

Testing *your* front-end
application to
production



Smoke Test 🔥 🔥



Bassam Ismail

Engineering Manager at Axelerant



skippednote

```
skippednote@xps: ~  
→ npx skippednote  
npx: installed 23 in 4.821s
```

Bassam Ismail (skippednote)
Engineering Manager at Axlerant.
RxJS / TypeScript / React / Angular / Testing

Web: <https://skippednote.dev>
Work: <https://www.axelerant.com>
GitHub: <https://github.com/skippednote>
Twitter: <https://twitter.com/skippednote>
LinkedIn: <https://linkedin.com/in/skippednote>

dd.skippednote.dev

Why should we test
our code?

To catch  before
our users can

Makes refactoring
possible

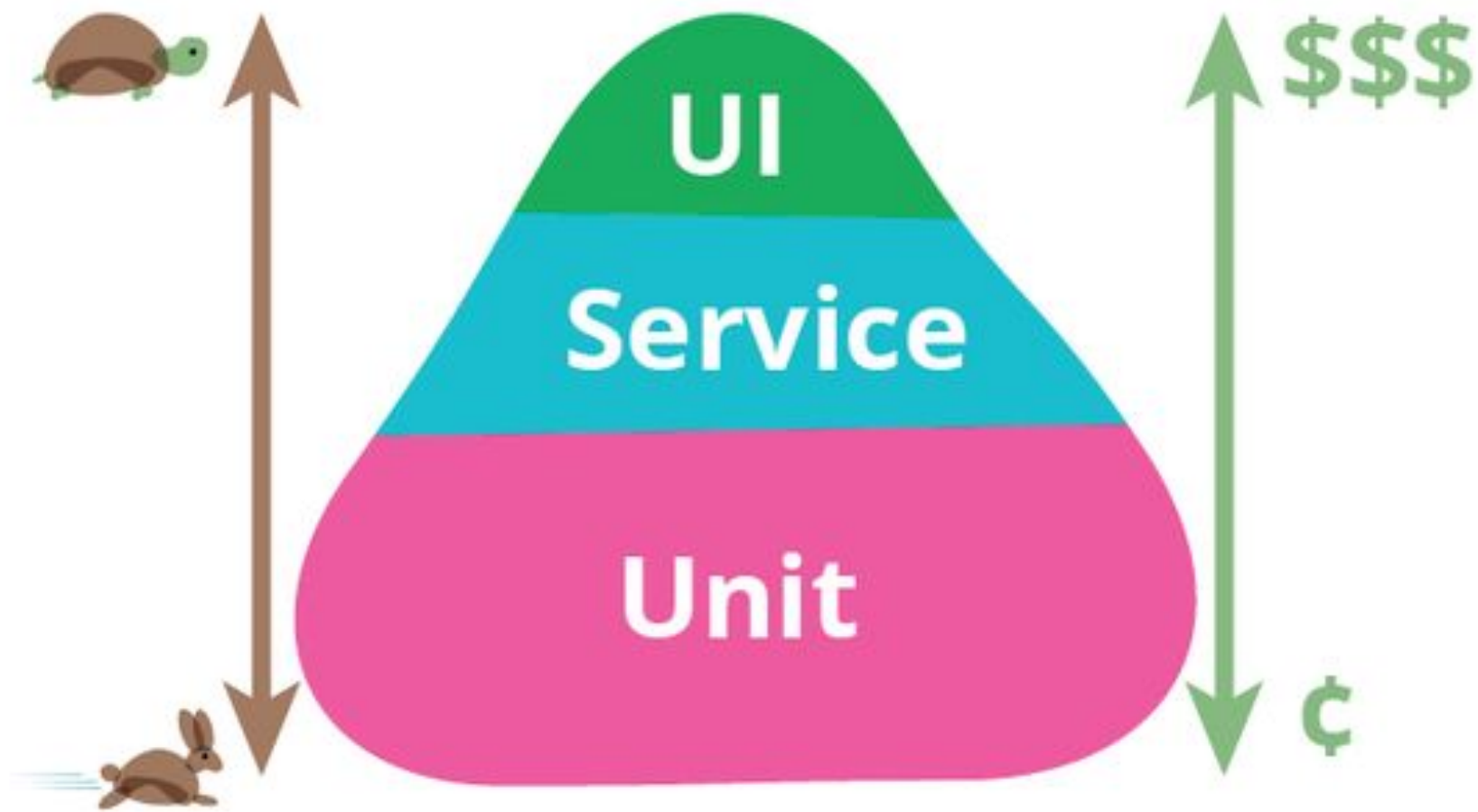
Helps write cleaner
code

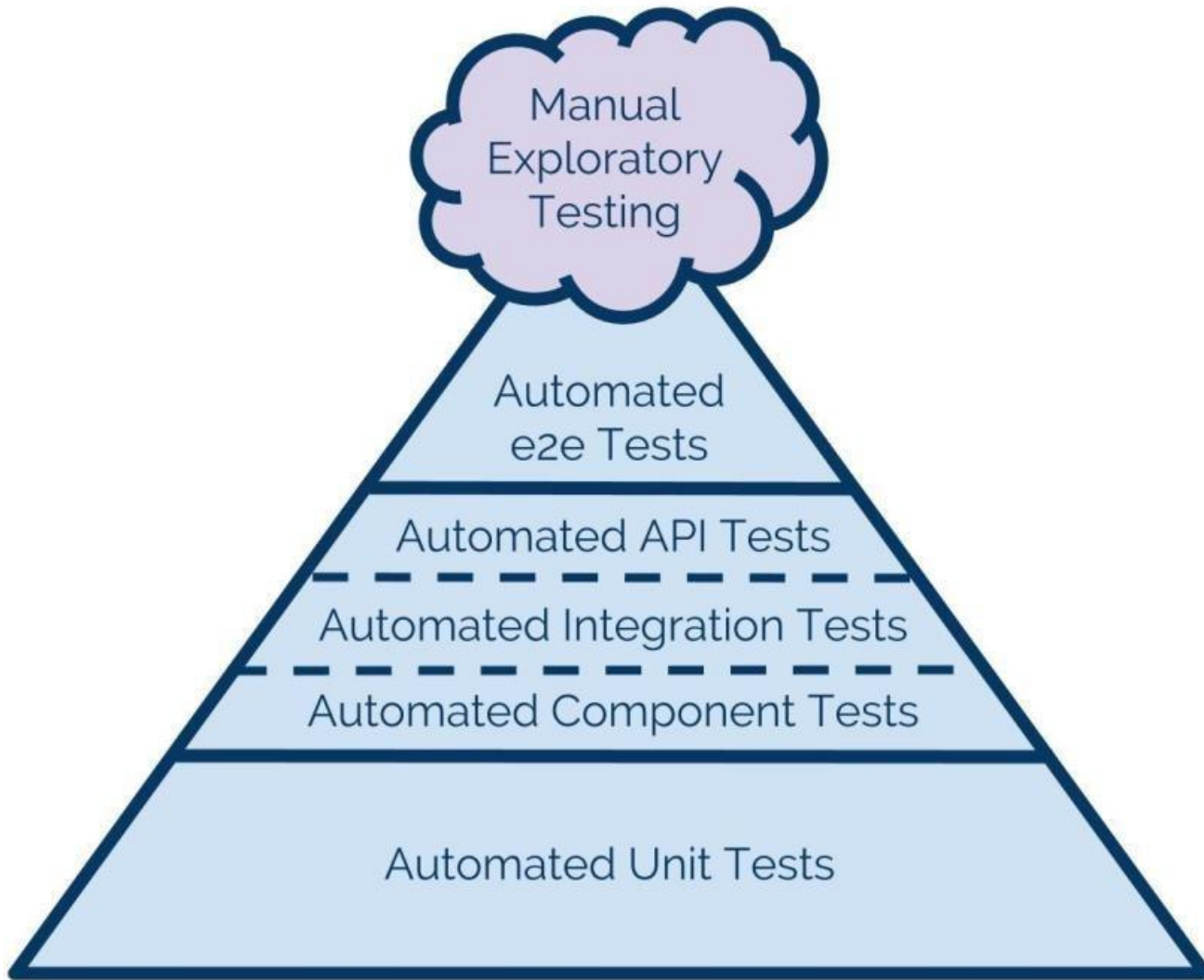
Acts as executable
documentation

Because all of us aren't
a 10x engineer

What should we
test in our code?







Manual
Exploratory
Testing

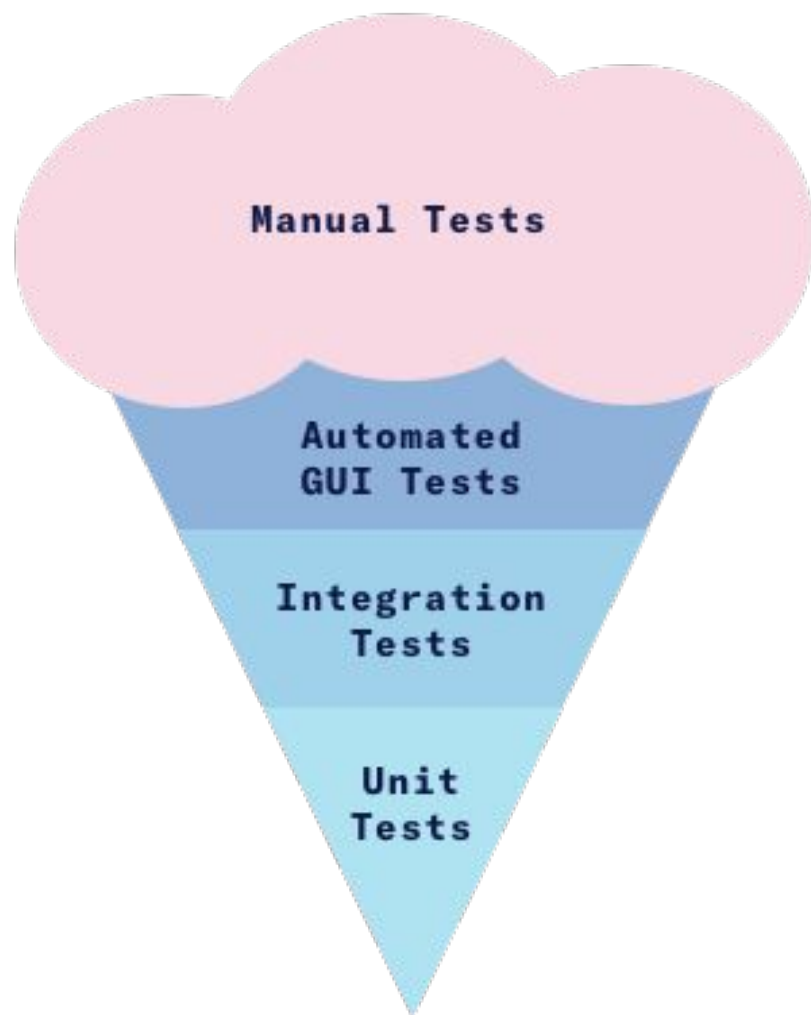
Automated
e2e Tests

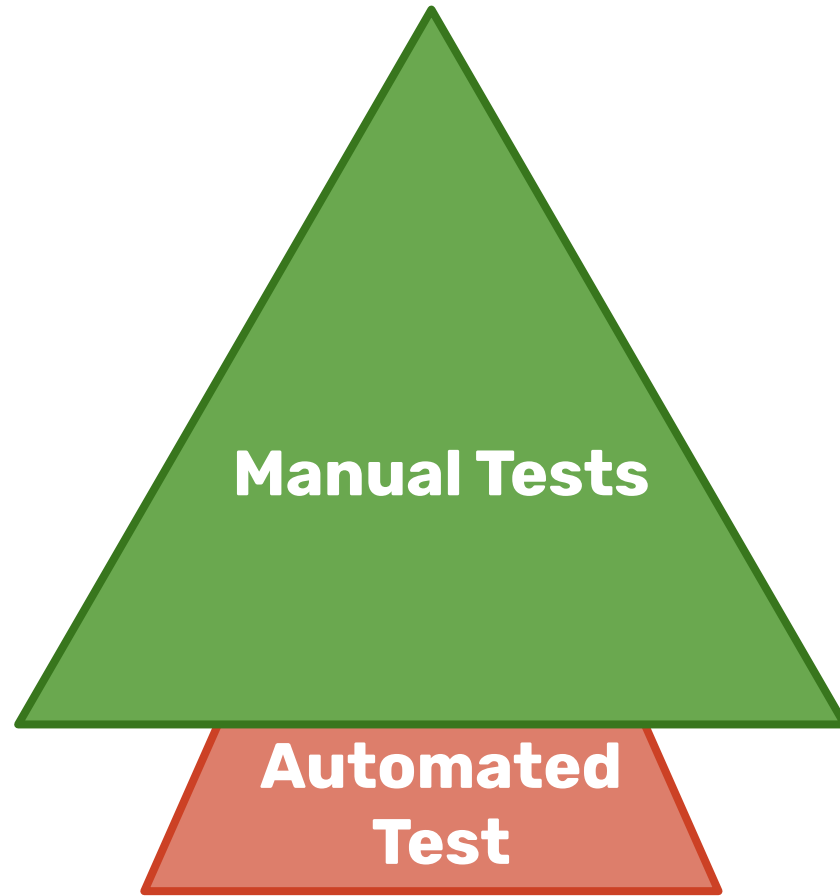
Automated API Tests

Automated Integration Tests

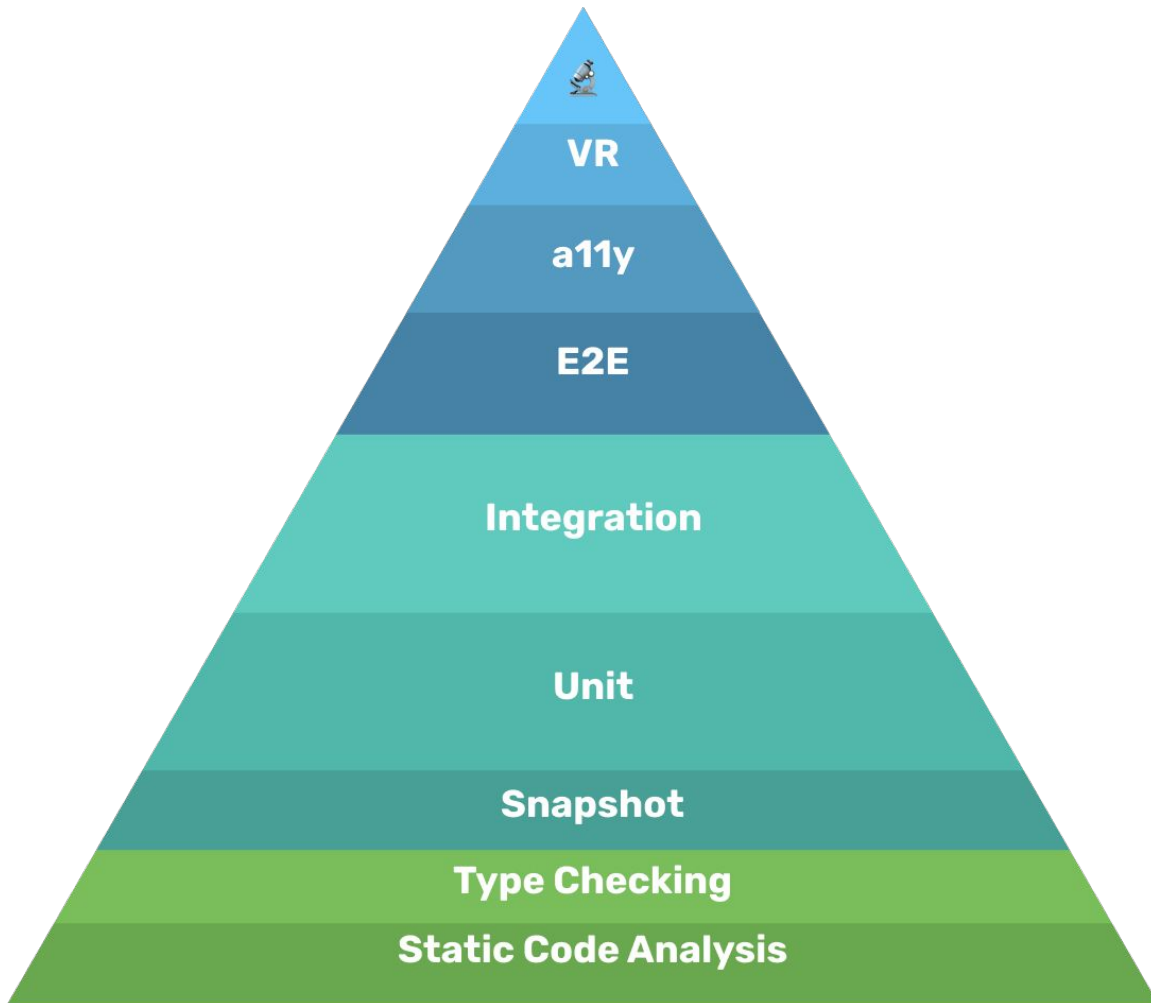
Automated Component Tests

Automated Unit Tests









Jargon



Test



```
it('can add two numbers', () => {  
  // setup  
  const add = Math.add()  
  
  // invocation  
  const total = add(23, 42, 12)  
  
  // assertion  
  expect(total).toEqual(77)  
})
```

Test Suite

```
describe('Math', () => {
  it('can add two numbers', () => {
    const add = Math.add()
    const total = add(23, 42, 12)
    expect(total).toEqual(77)
  })

  it('can subtract two numbers', () => {
    const subtract = Math.subtract()
    const total = subtract(23, 42, 12)
    expect(total).toEqual(-31)
  })

  it('can multiply two number', () => {
    const multiply = Math.multiply()
    const total = multiply(23, 42, 12)
    expect(total).toEqual(11592)
  })
})
```

Dummy / Fake



```
it('can filter by category', async () => {
  const sessions = [
    { title: 'hook', category: 'react' },
    { title: 'dependency injection', category: 'angular' },
    { title: 'virtual machine', category: 'vue' }
  ]

  const reactPosts = getByCategory(sessions, 'react')

  expect(reactPosts.length).toEqual(1)
  expect(reactPosts[0].title).toContain('hook')
})
```

Mocks



```
import { API as MockAPI } from "../..api";
jest.mock("../..api.js");

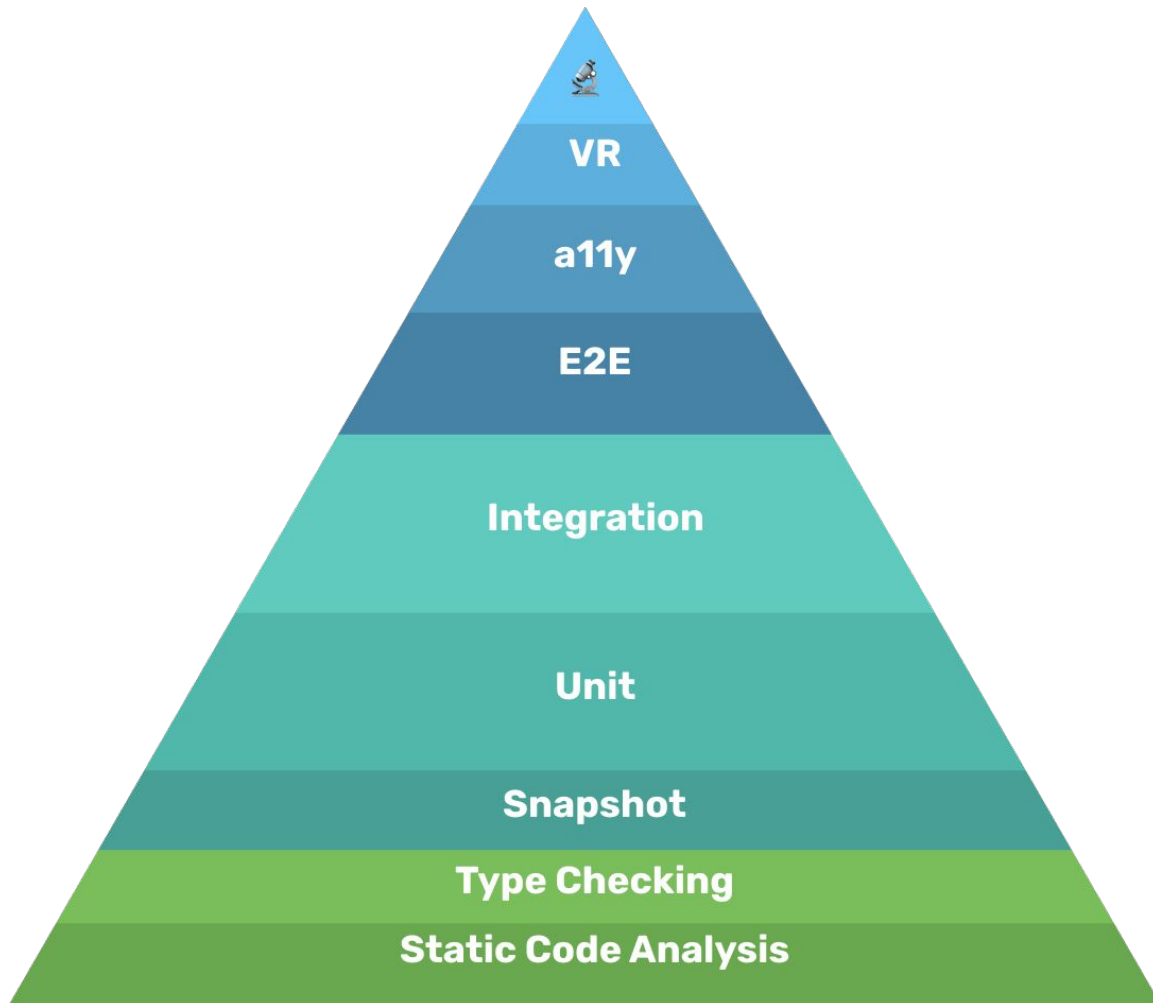
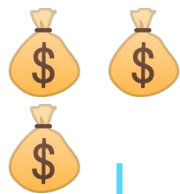
it('calls the API', async () => {
  const sessions = api.getSessions('42');

  expect(MockAPI.getSessions).toHaveBeenCalled();
  expect(MockAPI.getSessions).toHaveBeenCalledWith('42')
})
```

```

// src/__mocks__/api.js
export const API = {
  getSession: jest.fn(id => {
    if (id) {
      return { title: 'hooks', category: 'react' }
    } else {
      return [
        { title: 'hooks', category: 'react' },
        { title: 'dependency injection', category: 'angular' },
        { title: 'virtual machine', category: 'vue' },
      ]
    }
  })
}

```

Static Code Analysis





JS static-code.js ●



```
1  typeof "decoupled" === "strng";  
2  
3  if (1 * "a" === NaN) {  
4      true;  
5  }  
6  
7  JSON(data);  
8
```

Static Code Analysis

- easy to achieve 100% coverage
- runs automatically in your IDE/Editor
- helps reduce language grammar and syntax error

Type Checking

RE

TS





● typed-code.ts - fe - Visual Studio Code

TS typed-code.ts ●

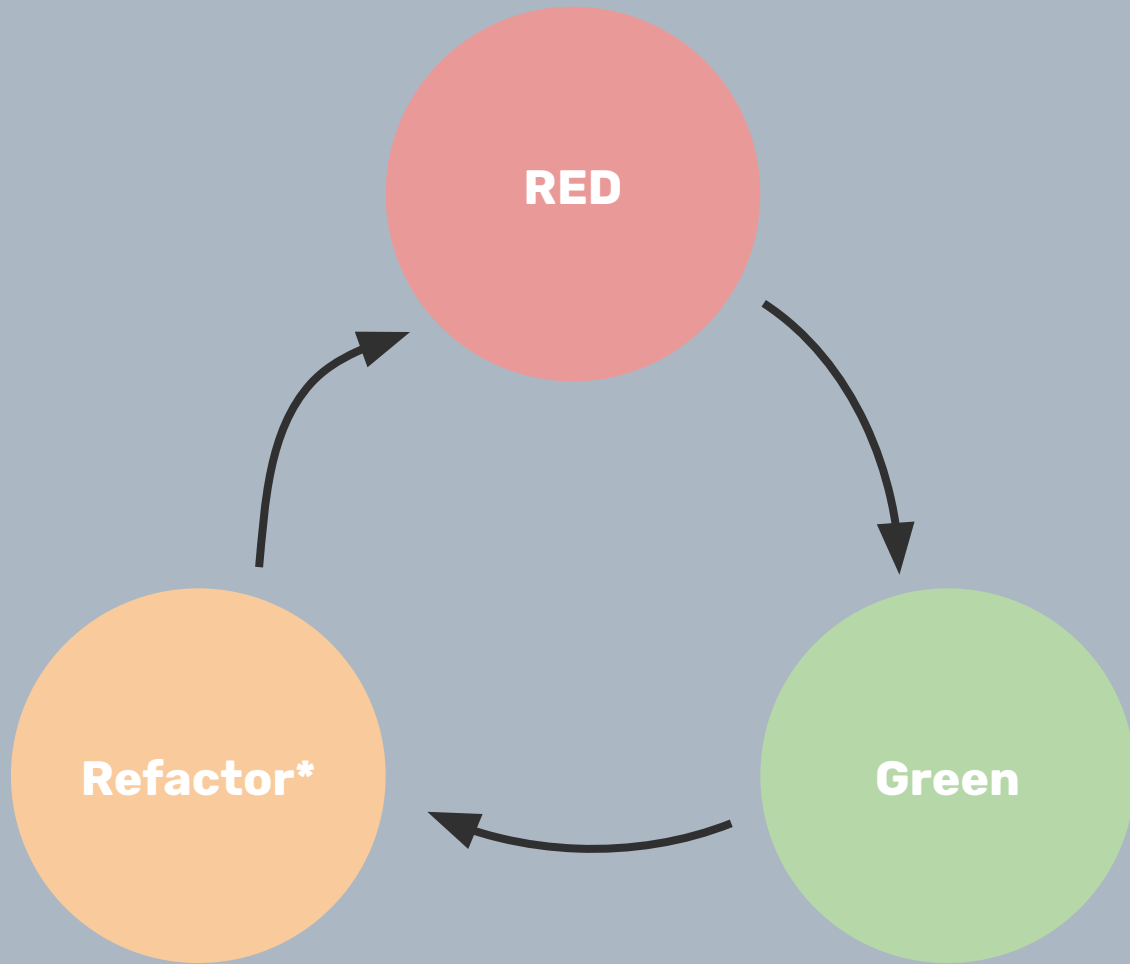


```
1  const input = document.querySelector("inpot");
2  input.value = "hello";
3
4  document.addEventListener("mousedown", e => {
5      e.clientX;
6      e.keyCode;
7  });
8
9
10
11
```


Type Checking

- helps prevent JavaScript gotchas
- provides great tooling and IDE support
- static code analysis beyond ESLint/TSLint

Unit Testing



Unit Testing

- lets us test in isolation
- easy to write and fast to run
- helps improve code design



use test specific
attributes for querying



```
it('toggles the filter', () => {  
  const filterButton = getByTestId('filter-button')  
  
  fireEvent.click(filterButton)  
  
  expect(getByPlaceholderText("Filter by input")).toBeInTheDocument();  
})
```



use regex when
querying text



```
it('toggles the filter', () => {  
  const filterButton = getByTestId('filter-button')  
  
  fireEvent.click(filterButton)  
  
  expect(getByPlaceholderText(/filter by input/).toBeInTheDocument());  
})
```


Snapshot Testing



```
exports[`renders the component 1`] = `  
  <div>  
    <button  
      class="button"  
      data-testid="show-filters"  
    >  
      <div  
        class="filter-label"  
      >  
        Show Filters  
      </div>  
    </button>  
  </div>  
`;  
;
```

Snapshot Testing

- easy to write and run
- warn against markup changes
- help capture current state of our application
- prevent UI bugs*

Integration Testing

2 Unit 0 Integration





Guillermo ▲ 

@rauchg



Write tests. Not too many. Mostly integration.

10:13 PM · Dec 10, 2016 from [San Francisco, CA](#) · [Twitter Web Client](#)

React App

localhost:3000

90%

Home

Create a session

Logout

Testing | Decoupled

Hide Filters

Filter by title

All Decoupled Drupal JavaScript and JAMstack

All 17 24

Before decoupling, understand JS: JavaScript for developers, let's learn this 25 minutes

room2 • 2019-07-17T00:12:12+00:00 • Decoupled Drupal

Bassam Ismail (Axelerant)

Bassam Ismail (Axelerant)

A node.js proxy between Drupal and your consumers 25 minutes

room2 • 2019-07-17T00:12:12+00:00 • Decoupled Drupal

Integration Testing

- Same benefits as Unit Testing
 - comparatively easy to write fast to run
 - helps improve code design
- Helps test components together
- Faster than E2E

E2E Testing

Session

✗ checks a11y on login page ⚠

TEST

- 1 VISIT /login
- (XHR) GET 200 /sockjs-node/info?t=156302...
- 2 - A11Y ERROR! color-contrast on 1 Node
- 3 - A11Y ERROR! label on 2 Nodes
- 4 - **ASSERT** 2 accessibility violations were detected: expected 2 to equal 0
- (XHR) GET 200 /8e56236b1321f3519301.hot-...
- (XHR) GET 200 /7d836a1bda2507edaba5.hot-...

AssertionError: 2 accessibility violations were detected: expected 2 to equal 0



Testing | Decoupled

Login

E2E Testing

- Tests like a real user
- Exposes browser inconsistencies
- Covers the whole application stack
- Slower than integration, however, no mock



extract **repeating** step
to **to** commands



```
Cypress.Commands.add("login", (email, password) => {  
  cy.visit("/login");  
  cy.get(":nth-child(1) > input").type("skippednote");  
  cy.get(":nth-child(2) > input").type("skippednote");  
  cy.get("form > .button").click();  
});
```

a11y Testing



Stefan Judis

@stefanjudis



The sad state of the internet.

[@webaim](#) analyzed the top 1,000,000 home pages and 97.8% included detectable [#a11y](#) errors.

Top ones:


- low contrast
- missing alt attr
- empty links
- missing form labels

webaim.org/projects/milli...

9:17 PM · Mar 19, 2019 · [Twitter Web App](#)



a11y Testing

- makes application accessible
- meet legal implications 
- write good and semantic HTML code
- become a better human being

Visual Regression Testing

React App

localhost:3000

- [Home](#)
- [Create a session](#)
- [Logout](#)

Testing | Decoupled

Hide Filters

Filter by title

All Decoupled Drupal JavaScript and JAMstack

All 17 24

[Before decoupling, understand JS: JavaScript for developers, let's learn this 25 minutes](#)

room2

-

2019-07-17T00:12:12+00:00

-

Decoupled Drupal

- Bassam Ismail (Axelerant)





You have used 1% of your free snapshots this month.

[More Info](#)



Build 2 i

All changes approved automatically on this branch

4 days ago

Baseline from **master**

Build 1 - 4 days ago

Changes from **master**

4 days ago

visual regression checks if the UI looks as expected



1280px



Auto-Approved

[Home](#)

Testing Decoupled

[Login](#)

Show Filters

E2E test 50 minutes

room3 • 2019-07-17T10:15:00+00:00 • JavaScript and JAMstack

Bassam Ismail (Axelerant)

E2E test 50 minutes

room3 • 2019-07-17T10:15:00+00:00 • JavaScript and JAMstack

Bassam Ismail (Axelerant)

[Home](#)

Testing ~~Decoupled~~

[Login](#)

Show Filters

E2E test 50 minutes

room3 • 2019-07-17T10:15:00+00:00 • JavaScript and JAMstack

Bassam Ismail (Axelerant)

E2E test 50 minutes

room3 • 2019-07-17T10:15:00+00:00 • JavaScript and JAMstack

Bassam Ismail (Axelerant)



Visual Regression Testing

- Helps verify the UI isn't broken
- Find cross browser compatibility bugs

Monitoring




Monitoring

- quick turn over between bug detection and fix
- prevent revenue loss





More  === Less



Refactoring Cleanup

Questions?

Thank you

dd.skippdenote.dev
@skippednote