



Cypress

Cypress Overview

- “Fast, easy and reliable testing for anything that runs in a browser”
 - <https://www.cypress.io/>
- Supports end-to-end (E2E) tests
- Only runs in Chrome, no other browsers
- Can install Cypress plugin through Vue CLI
 - creates `tests/e2e/specs/test.js` containing an example test
 - creates `tests/e2e/plugins/index.js` that configures the Cypress plugin
- Tests can be written in any `.js` file under `e2e/specs`
- To run tests
 - enter `npm run test:e2e`
 - click “Run all specs” button

Cypress UI

- Show a screenshot of tests that have run
- Passing tests are green, failing are red
- Can click on a test to rerun it?
- Can see output of tests?
- Is watch mode automatic to rerun tests on code changes?
- Implement more tests in Know-Your-CIO-survey

Writing Tests

- Use CSS selectors to find page elements to test
- Can use `data-` attributes
- Sample test

```
// https://docs.cypress.io/api/introduction/api.html  
  
describe('My First Test', () => {  
  it('Visits the app root url', () => {  
    cy.visit('/');  
    cy.contains('h1', 'Welcome to Your Vue.js App');  
  });  
});
```

- Tests use familiar methods found in other test libraries like Jest
 - `describe`, `before`, `beforeEach`, `after`, `afterEach`, `it`, ...
- `cy` is a global variable

Cypress Commands

- Official docs at <https://docs.cypress.io/api/api/table-of-contents.html>
- Highlights
 - `cy.visit(url, options);`
 - `const element = cy.get(selector);`
 - for example, a selector for an element with a given data attribute is `'[data-name]'`
 - `contains`

Viewport

- Can set the viewport size
 - specific width and height
 - device name
- Example
 - `cy.viewport('ipad-2');`

Cypress Assertions

- Supports BDD, TDD, Chai, and Sinon assertions
- Common Examples - **VERIFY THESE!**
 - value - `expect(element.value).to.be(value);`
 - text content - `expect(element).to.contain(text);`
 - text content approach 2? - `element.contains(text);`
 - length - `expect(array.length).to.be(value);`
 - class - `expect(element).to.have.class(className);`
 - visibility - `expect(element).to.be.visible;`
 - enabled - `expect(element).to.be.enabled;`
 - existence - `expect(element).to.exist;`
 - checked - `expect(element).to.be.checked;`
 - include - `expect(array).to.include(value);`
- Can add `.not` before `.to`

Cypress Events

Cypress Utilities