

25. End-to-End Testing with Cypress

[Github Repository](#) | [Vimeo Video Recording](#)

Topics to cover

- [x] 1. End-to-End (A-Z) Testing
- [x] 2. Jest vs Cypress
- [x] 3. Install and Configure Cypress
- [x] 4. Design End-to-End Tests with Cypress

Jest vs Cypress

- Jest
 - Command line test runner
 - Based around testing assertions
 - Used for unit and integration testing (mostly)
- Cypress
 - Runs its own browser to execute the tests in
 - Performs operations and interacts with the site the way that a user would (eg. typing into input fields, clicking on buttons)
 - Used for integration and E2E testing (mostly)

Install and Configure Cypress

- Cypress can be installed locally to the project (as a dev dependency) or globally on your OS

```
npm install -g cypress
npm install --save-dev cypress
```

- Use the `open` command to start Cypress running

```
# global installation
cypress open

# local installation
node_modules/.bin/cypress open
```

- Add a script to `package.json` for a quick way to start Cypress

```
"cypress": "node_modules/.bin/cypress open"
```

```
npm run cypress
```

- We use the `cypress.json` file in the main directory to configure Cypress

```
{
  "baseUrl": "http://localhost:3000",
  "viewportWidth": 1280,
  "viewportHeight": 1200
}
```

- `baseUrl` tells Cypress where our application is hosted and what port it's listening on
- `viewportWidth` and `viewportHeight` specify the dimensions for Cypress' browser to use
- Feel free to change the width and height values if developing for a mobile-first site

Some Old Friends

- Cypress is built on top of Mocha and uses Chai assertions
- Cypress comes with jQuery installed (accessible with `Cypress.$()` ;)

Useful Links

- [Cypress Docs](#)
- [Cypress Best Practices \(Official\)](#)

[View on Compass](#) | [Leave Feedback](#)