



Workshop Cypress



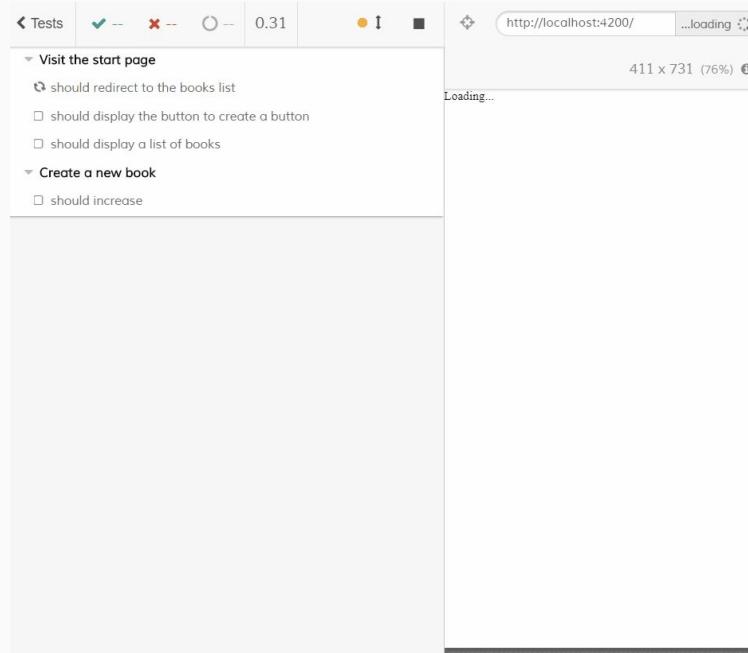


Fast, easy and reliable testing.

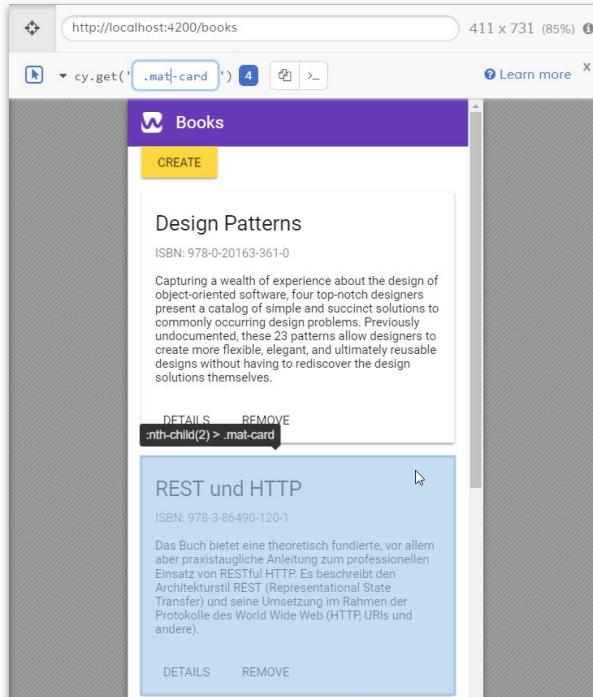
Benefits of cypress

- Faster test runs
- Intuitive tooling for Non-Developers (e.g. Time Travel)
- No need to handle timeouts querying DOM
- Independent from Front-End technologies

Cypress Test Runner



Cypress Test Runner



Selector Playground

- Analyze your App as your Tests run.
- Find CSS-Selectors with ease.

Task

Install cypress



If you know **protractor** you will love **cypress**.

Write UI-Tests

- You do not need to learn a new syntax.
- Cypress' tests are structured like tests written with  JASMINE.
- Cypress' API looks like PROTRACTOR's API.
- Cypress has its own implementation of *Promises*.
 - Currently *async/await does not work*

Hooks | Old friends

```
describe('Hooks', () => {  
  before(() => { /* runs once before all tests in the block */ })  
  
  after(() => { /* runs once after all tests in the block */ })  
  
  beforeEach(() => { /* runs before each test in the block */ })  
  
  afterEach(() => { /* runs after each test in the block */ })  
})
```

Structure a test

```
describe('Visiting the start page', () => {  
  context('is working', () => {  
    it('should redirect to the books list');  
  });  
});
```

Scope test runs

.only()	Applied to a <code>describe</code> , <code>it</code> or <code>context</code> it only runs that target.
.skip()	Excludes the specified target from the test run.

Example

```
describe.only('Visiting the start page', () => {  
  it.skip('should skip this test');  
  it('should run this test');  
});
```

Control your App

```
// Open the page
cy.visit('http://localhost:4200');

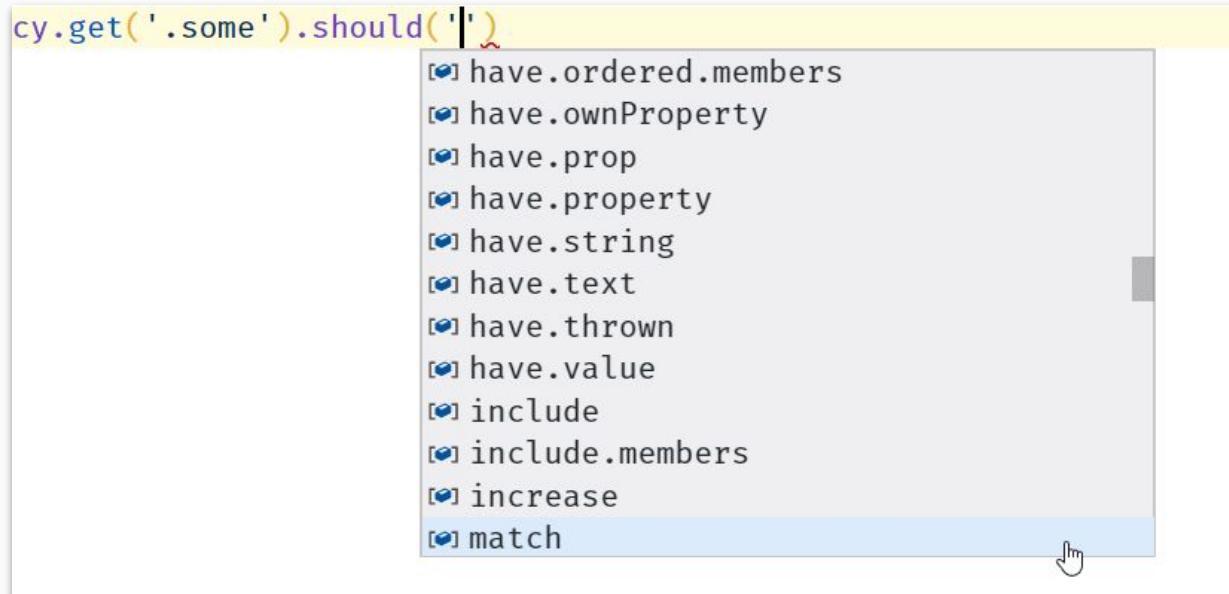
// Query DOM
cy.get('.mat-raised-button')
```

Setup expectations

```
// Check Content
cy.get('.mat-raised-button').contains(/create/i);

// Check property of element
cy.get('.mat-card').should('have.length.greaterThan', 1);
```

All matchers are typed



A screenshot of a code editor showing a dropdown menu of matcher functions. The code being typed is `cy.get('.some').should('|')`. The dropdown menu lists the following matchers:

- have.ordered.members
- have.ownProperty
- have.prop
- have.property
- have.string
- have.text
- have.thrown
- have.value
- include
- include.members
- increase
- match

The word `match` is highlighted in blue, indicating it is the next available option to complete the matcher.

Please refer to the documentation



<https://docs.cypress.io>

Task

Query DOM with cypress



Interact with UI elements

- [.click\(\)](#)
- [.dblclick\(\)](#)
- [.type\(\)](#)
- [.clear\(\)](#)
- [.check\(\)](#)
- [.uncheck\(\)](#)
- [.select\(\)](#)
- [.trigger\(\)](#)

Example

```
// Click a button
cy.get('.mat-raised-button').click();

// Fill in a value into a input control
cy.get('[formcontrolname="isbn"]').type('948234728');
```

Example - Query and Interact

```
cy
  .get('.mat-raised-button')
    .contains(/create/i)
  .click();
```

Aliases | Reuse Queries

```
// Define
cy.get('.mat-card').as('books')

// Use
cy.get('@books')
```

Examine Aliases

```
// cy.get('.mat-card').as('books');

let countBefore = 0;
cy.get('@books').then(books => (countBefore = books.length));
```

Task Page Interactions



Perform Network Requests

API

```
cy.request(url);
cy.request(url, body);
cy.request(method, url);
cy.request(method, url, body);
cy.request(options);
```

Sample

```
cy.request('POST', 'http://localhost:4730/books/', {});
```

Task

Simplify Test Clean Up



Stub XHR-Backend

✓ Benefits

- Easy control of response bodies, status, and headers
- Can force responses to take longer to simulate network delay
- No code changes to your server or client code
- Fast, < 20ms response times

<https://docs.cypress.io/guides/guides/network-requests.html#Stub-Responses>

Stub XHR-Backend

✖️ Downsides

- No guarantee your stubbed responses match the actual data the server sends
- No test coverage on some server endpoints
- Not as useful if you're using traditional server side HTML rendering

Stub XHR-Backend

➊ Suggested Use

- Use for the vast majority of tests
- Mix and match, typically have one true end to end test, and then stub the rest
- Perfect for JSON API's

Stub XHR-Backend

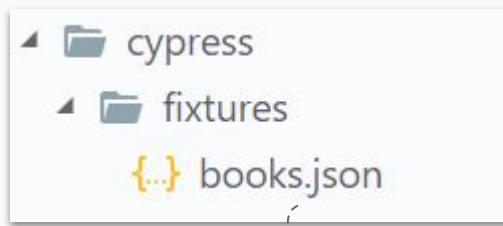
API

```
// spying and response stubbing
cy.intercept(url, staticResponse);
cy.intercept(method, url, staticResponse);
cy.intercept(routeMatcher, staticResponse);
cy.intercept(url, routeMatcher, staticResponse);
```

Sample

```
cy.intercept('http://url/resource', reponse);
cy.intercept('**/resource', reponse);
```

Fixtures | Organize Test Data



Sample

```
cy.intercept('http://localhost:4730/books', { fixture: 'books' });
```

Task Stub XHR

