Master Cypress in 15 minutes a day

Maurice de Beijer - @mauricedb

Course Goals

cypress

Course Goals

- Learn what Cypress is
  - And what it can do for you
- Understand what the best practices are
  - And what not to do
- See how Cypress differs from other End to End testing tools
  - Like Selenium, Nightwatch.js etc.

cypress

Why this course?

- Cypress is not hard to use most of the time
  - Not knowing the best practices can trip you up
- Going beyond the basics can be tough

cypress

15 minutes

- Can I really do this in 15 minutes a day?

cypress

See you in the next video

cypress

What is Cypress

And why use it?

What is Cypress

- Cypress is a front end testing tool
  - Built for the modern web
- Tests are easy to write
  - Using simple JavaScript
- Time Travel runner lets you inspect test at each step
  - Click on a step to see the state of the browser
- Automatic waiting for elements
  - Makes tests more resilient
- Video recording of running test
  - Helps debugging failing tests on the CI server

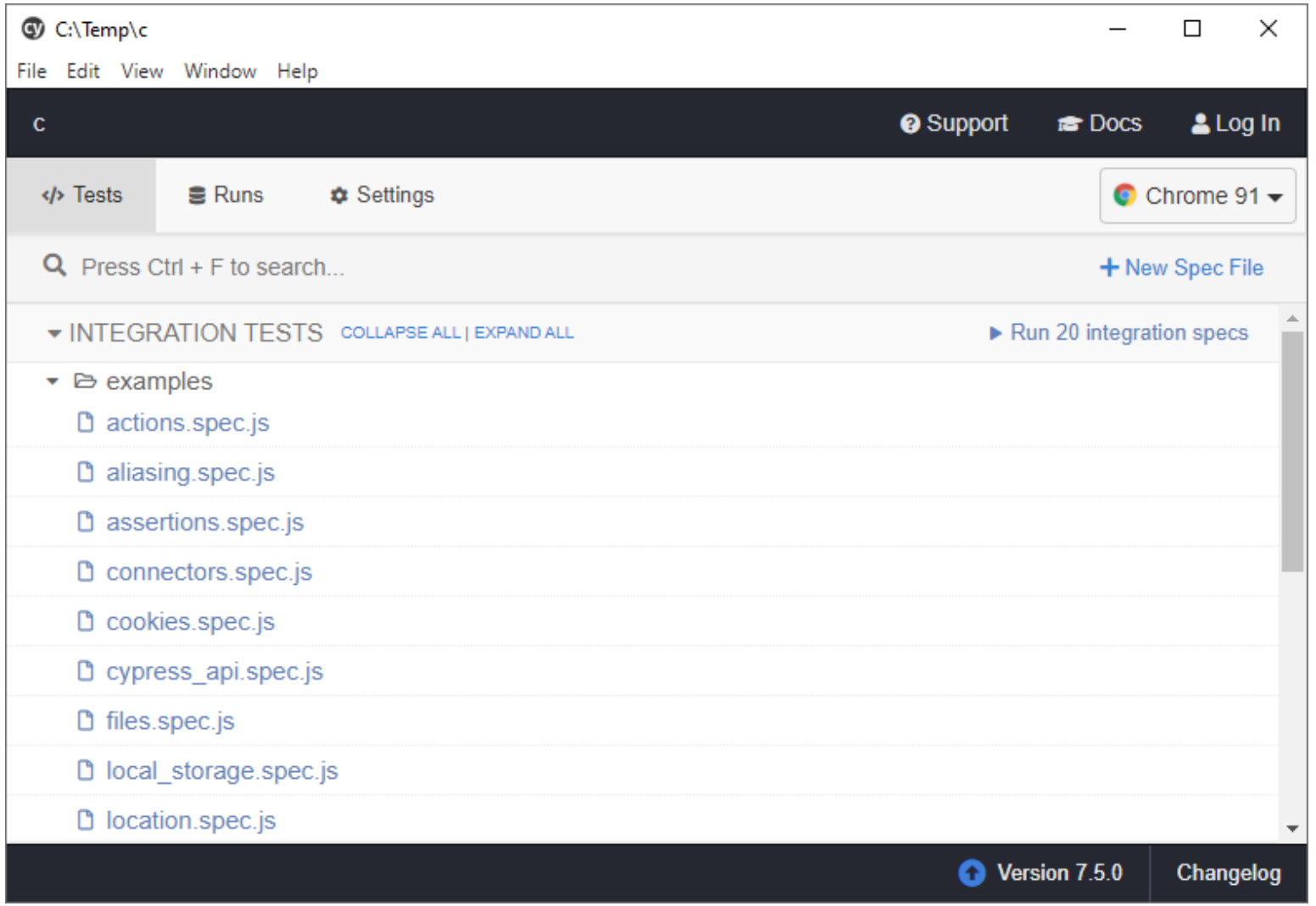
cypress

What is Cypress

- Lets you fake the network
  - Or use the real backend as appropriate
- Automatically retry
  - Run flaky tests multiple times
- Many standard and 3rd party extensions
  - Or write your own in JavaScript
- Can run in Docker containers
  - Consistent cross platform behavior
- Runs inside the browser
  - Not based on Selenium

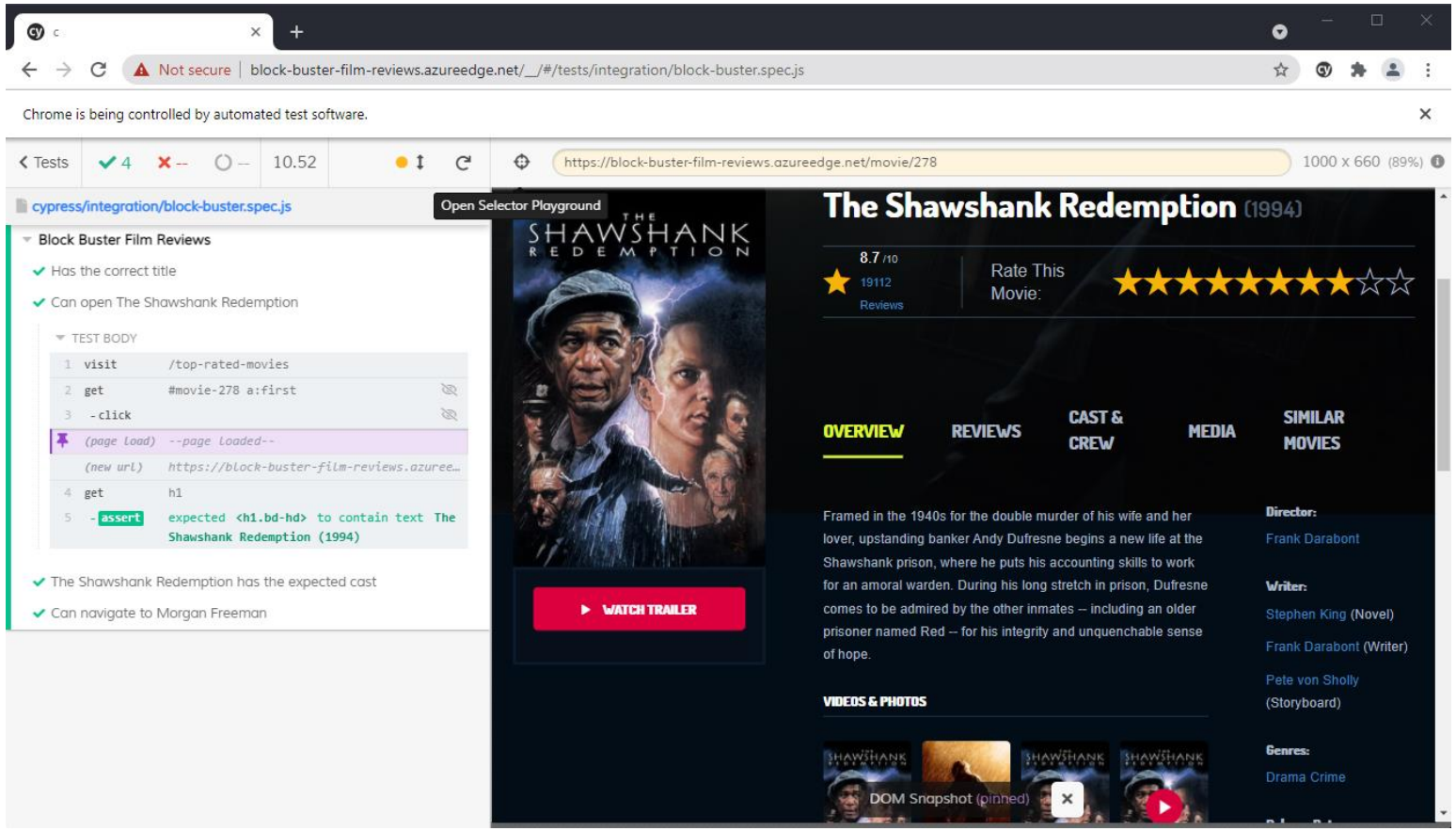
cypress

Interactive Test Runner

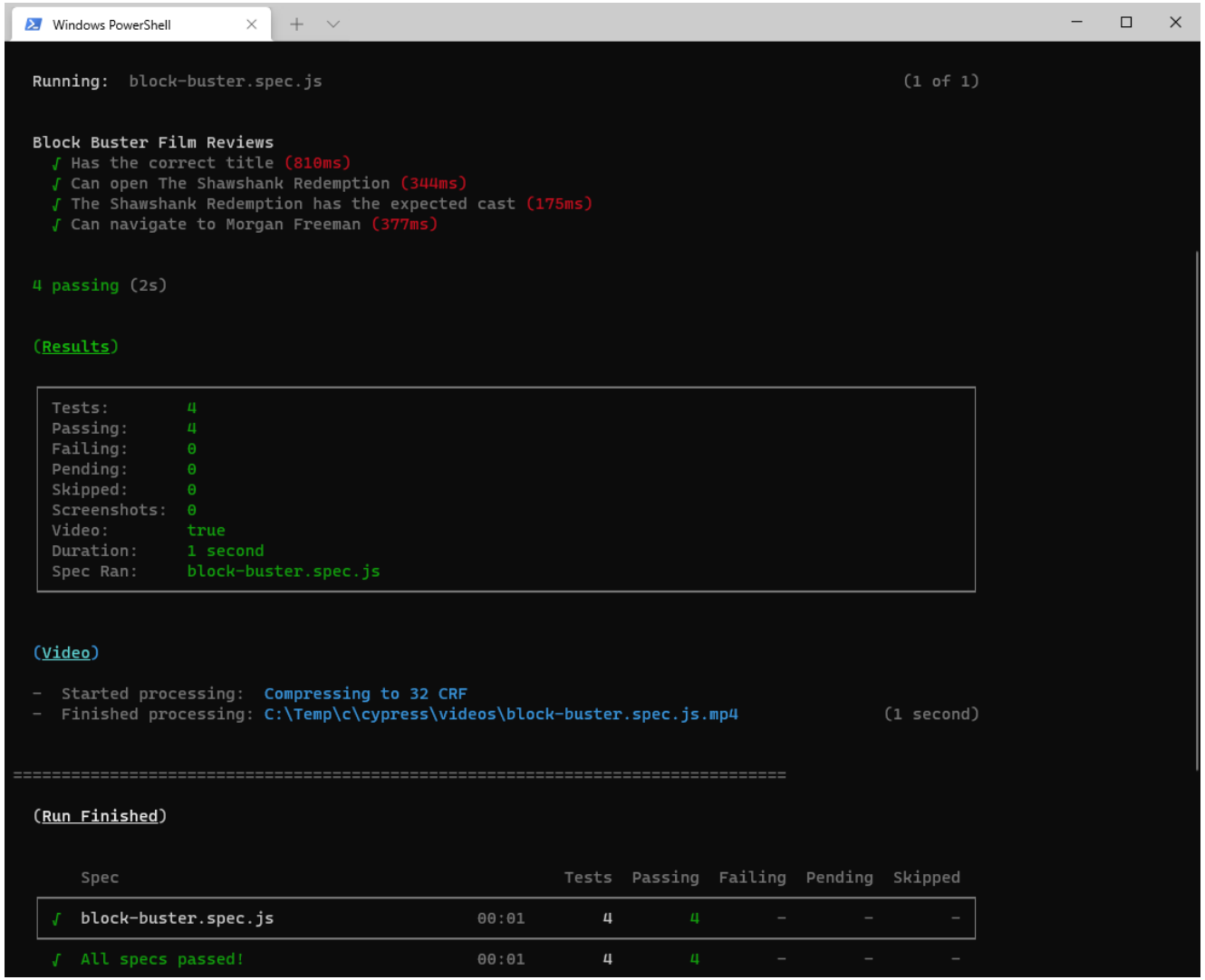


cypress

Interactive Test Runner



Command Line Interface



Pros and Cons of Cypress

Pros

- Easy to get started
- Fast execution
- Automatic waiting on elements
- Access to elements outside the DOM
- Support for FireFox and Chromium based browsers
- No driver dependencies
- Time travel and debuggability
- Uses the Mocha test framework
- Open source
- Mostly free

Cons

- Only JavaScript or TypeScript
- No support for Safari
- Only a single bowser
- Only a single tab

cypress

See you in the next video

cypress

What is end to end testing

cypress

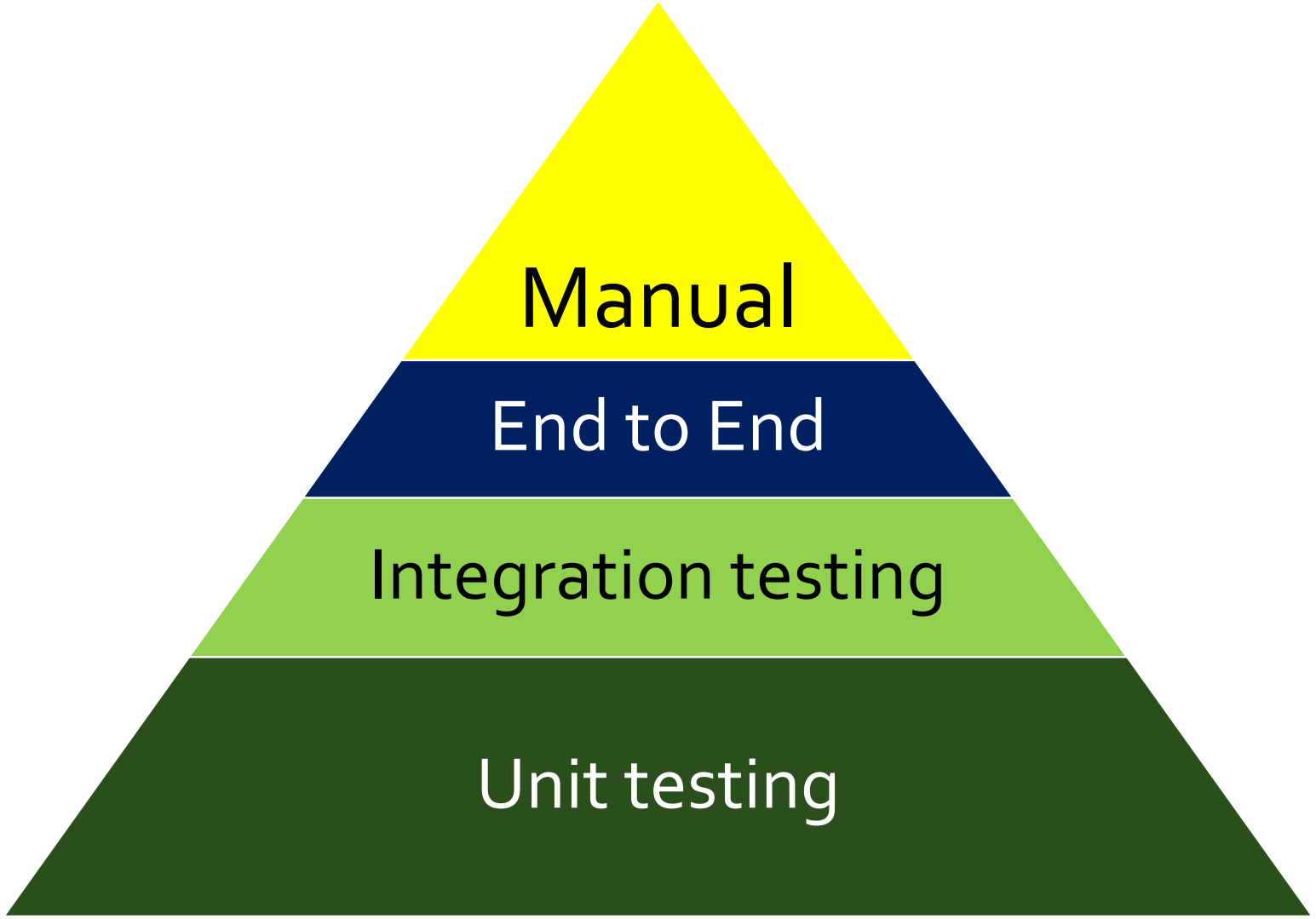
What is testing

"Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test."

-- Wikipedia --

cypress

The traditional testing pyramid



cypress

Unit Testing & Code Coverage

```
----------------|----------|----------|----------|----------|----------------|
File            |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
----------------|----------|----------|----------|----------|----------------|
All files       |    98.92 |    94.36 |    99.49 |      100 |                |
 yargs          |    99.17 |    93.95 |      100 |      100 |                |
  index.js      |      100 |      100 |      100 |      100 |                |
  yargs.js      |    99.15 |    93.86 |      100 |      100 |                |
 yargs/lib      |     98.7 |    94.72 |    99.07 |      100 |                |
  command.js    |     99.1 |    98.51 |      100 |      100 |                |
  completion.js |      100 |    95.83 |      100 |      100 |                |
  obj-filter.js |     87.5 |    83.33 |    66.67 |      100 |                |
  usage.js      |    97.89 |    92.59 |      100 |      100 |                |
  validation.js |      100 |    95.56 |      100 |      100 |                |
----------------|----------|----------|----------|----------|----------------|
```

cypress

Both windows are fine

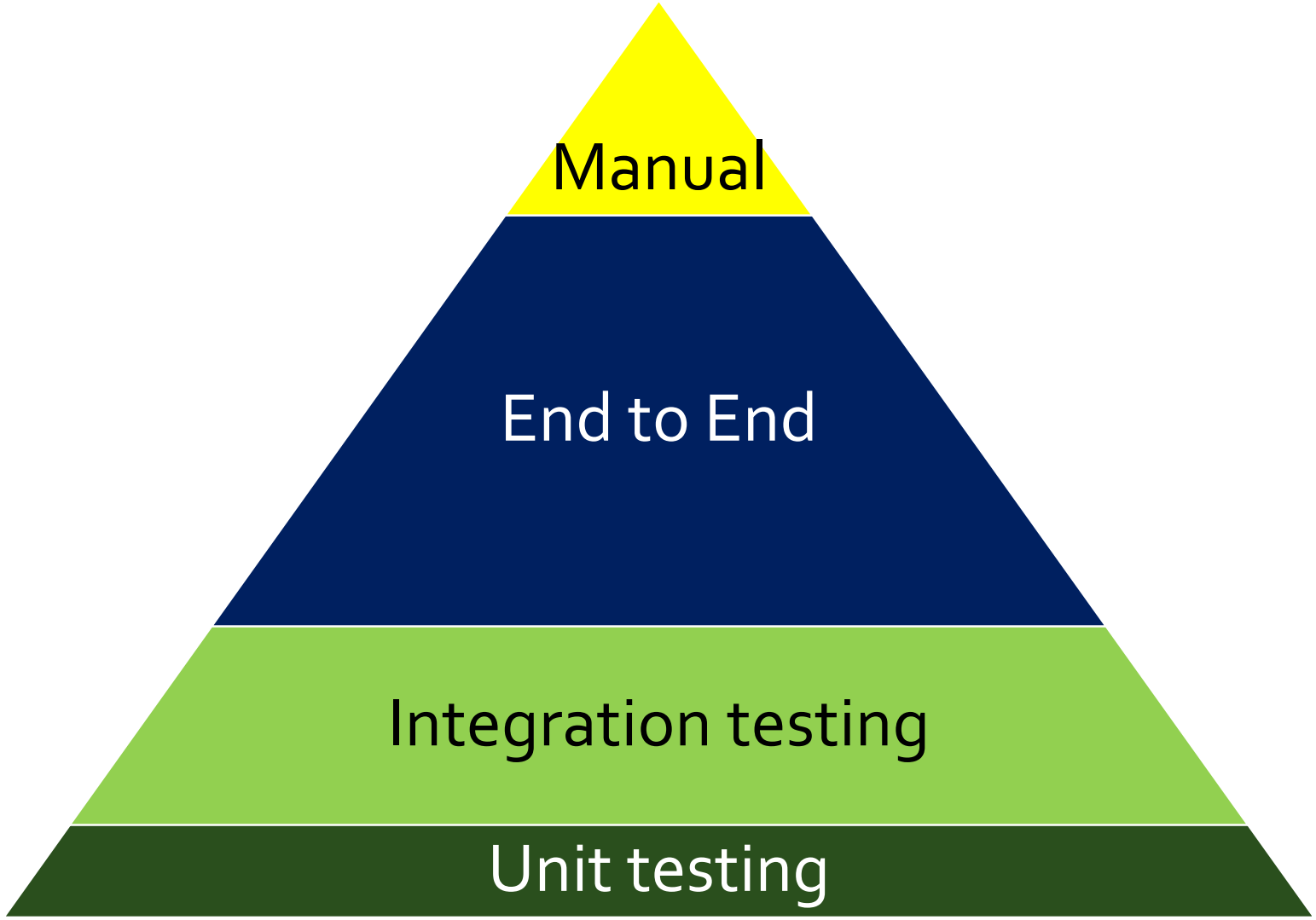
Source

cypress

A sturdy latch

Source

A better testing pyramid for the web



cypress

End-to-end Testing

"End-to-end testing is a methodology used to test whether the flow of an application is performing as designed from start to finish. The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various system components and systems."

-- Techopedia --

cypress

End-to-end Testing

- Test an application's workflow from beginning to end
  - Including AJAX requests to the backend
- Based on workflows the end user actually executes
  - Using the same user interface, browser, network database etc.

cypress

See you in the next video

cypress

Personal introduction

cypress

Maurice de Beijer

Independent software developer and trainer

cypress

The Netherlands

cypress

Happily married

cypress

IN SEARCH OF
SAILOR SPIRIT
SOUL
FOLLOW
TWINLIFE
SONG

Independent software developer & instructor

Since 1995

cypress

cypress

The React Newsletter

cypress

See you in the next video

cypress

Prerequisites

Install Node & NPM

Install the GitHub starter repository

cypress

Following Along

```
/// <reference types="cypress" />

context('Block Buster Film Reviews', ()=>{

  it('Has the correct title', () =>{
    cy.visit('/')
    cy.get('h1').should('contain.text', 'Block Buster Film Reviews')
  })

  it('Can open The Shawshank Redemption', () => {
    cy.visit('/top-rated-movies')
    cy.get('#movie-278 a:first').click()
    cy.get('h1').should('contain.text', 'The Shawshank Redemption (1994)')
  })

  it('The Shawshank Redemption has the expected cast', () => {
    cy.visit('/movie/278')
    cy.get('.col-md-8 > .mvcast-item > #cast-504 > p').should('have.text', '... Andy Dufresne')
    cy.get('.col-md-8 > .mvcast-item > #cast-192 > p').should('have.text', '... Ellis Boyd "Red" Redding')
    cy.get('.col-md-8 > .mvcast-item > #cast-192 > .cast-left').should('contain.text', 'Morgan Freeman')
  })

  it('Can navigate to Morgan Freeman', () => {
    cy.visit('/movie/278')
    cy.get('.col-md-8 > .mvcast-item > #cast-192 a').click()
    cy.get('h1').should('contain.text', 'Morgan Freeman')
  })
})
```

- Slides:
  - http://theproblemsolver.nl/master-cypress-in-15-minutes-a-day.pdf

cypress

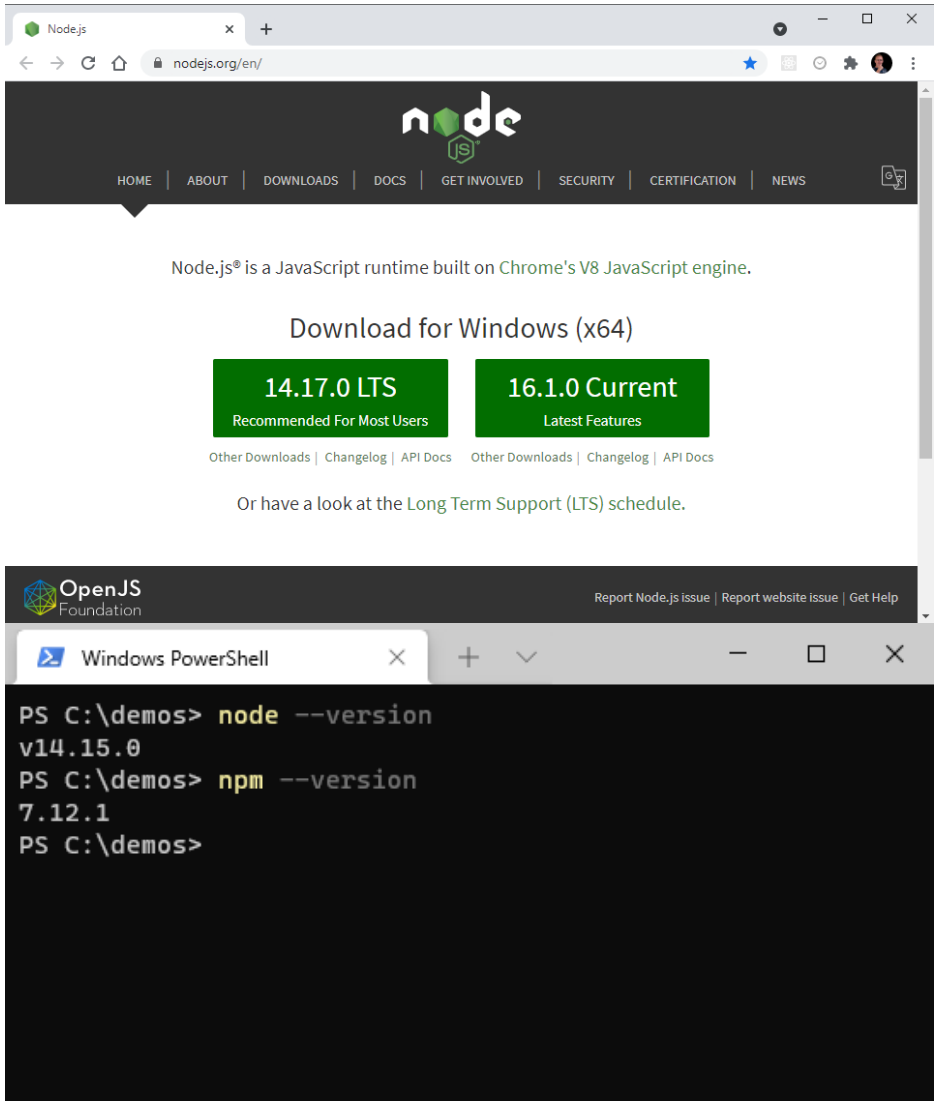
Type it out by hand?

"Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now!"

cypress

Install Node.js

- Minimal:
  - Node version 12



cypress

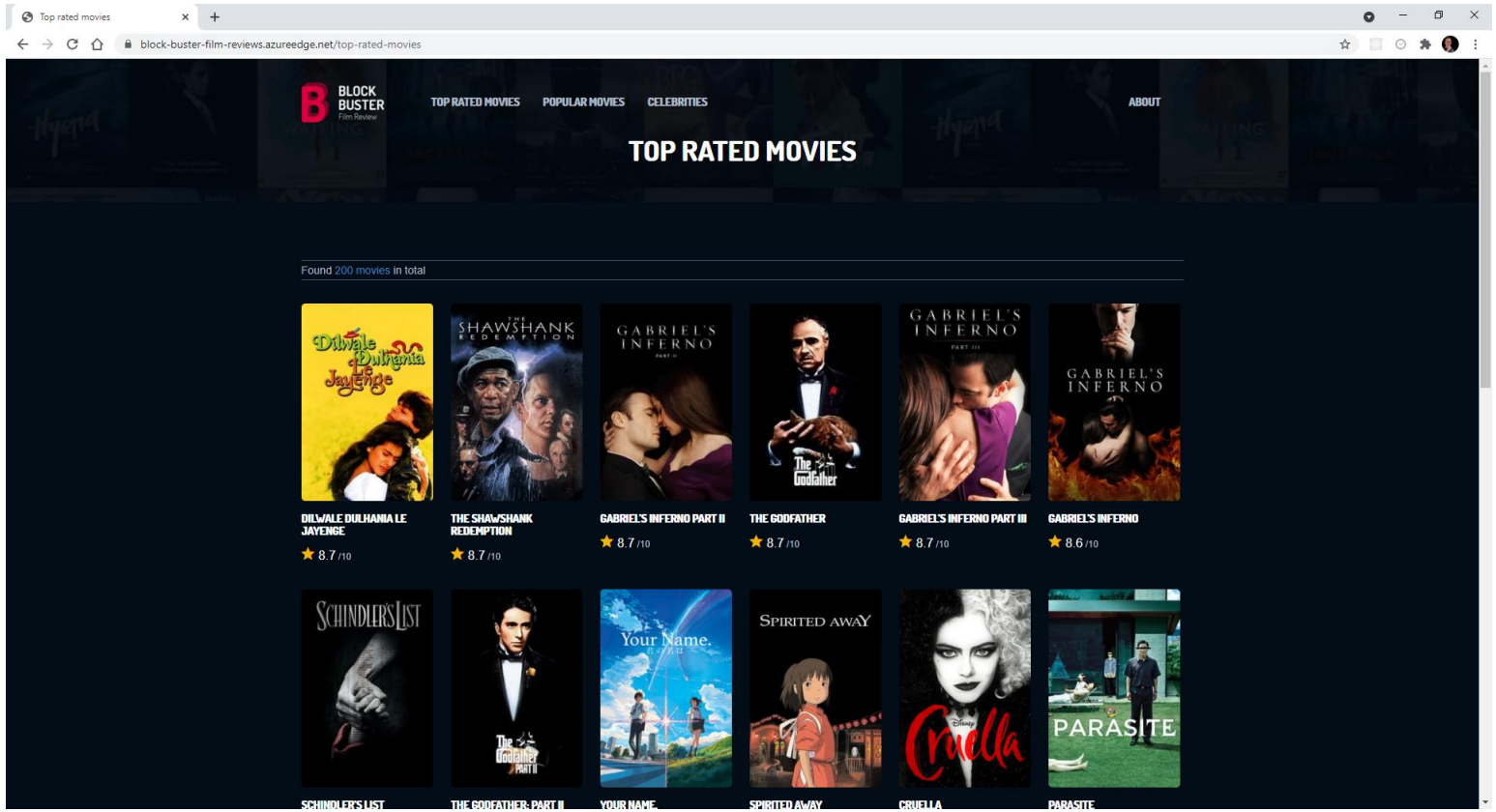
See you in the next video

cypress

Creating a first Cypress test

cypress

The Application



cypress

Opening Cypress

- Start Cypress using "npx cypress open"
  - This will automatically download and run Cypress
- The first time it will create the Cypress a folder structure
  - Including a number of example tests

cypress

Creating a first Cypress test

- Create a new spec file in the **/cypress/integration** folder
  - The spec file name must end with **spec.js**
- Call the **context()** function to create a group of specification
  - The **describe()** function is an alias for context()
- Call the **it()** function to create a End to End specification
  - Both **context()** and **it()** come from the Mocha test framework

cypress

Creating a first Cypress test

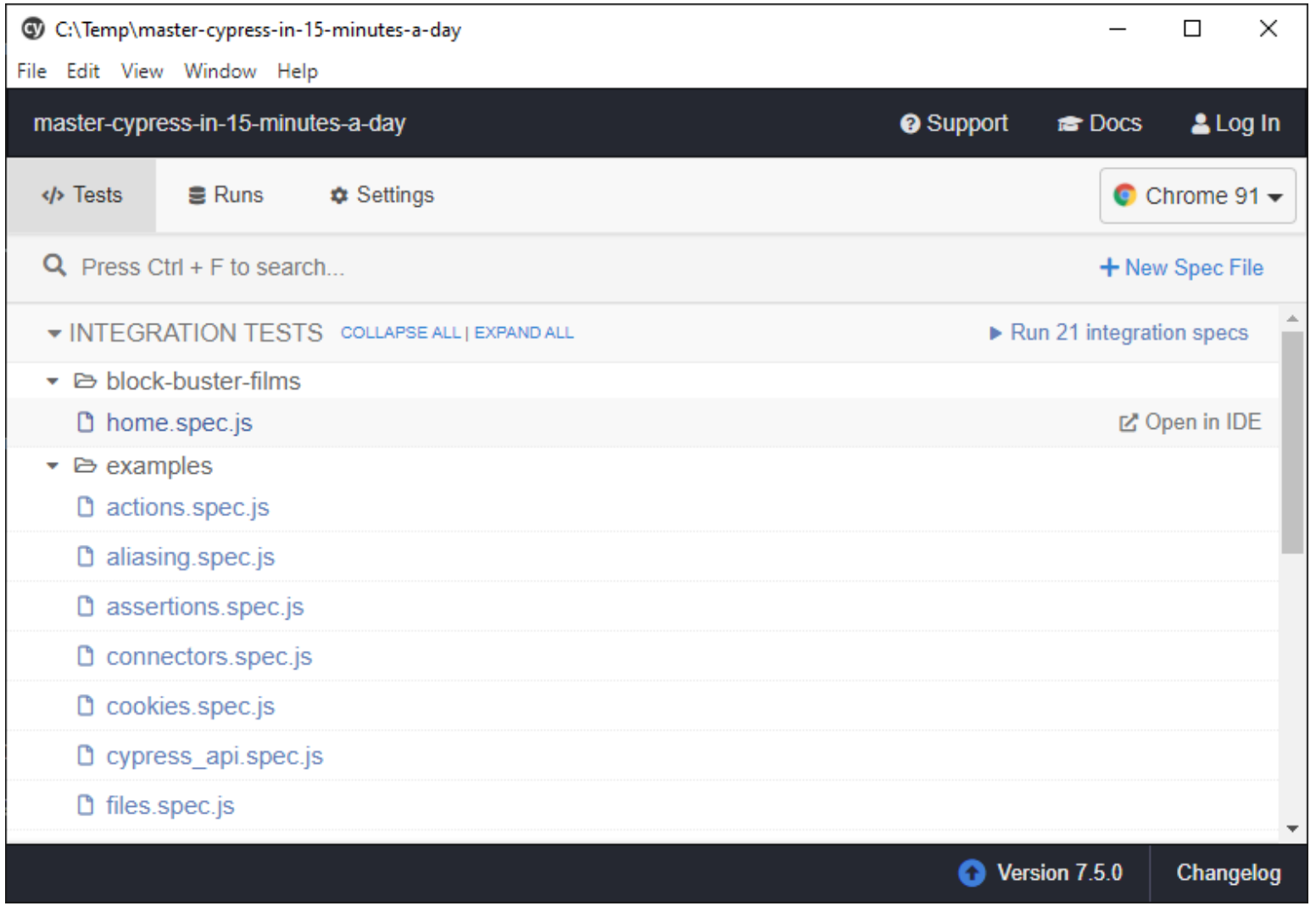
home.spec.js

cypress > integration > block-buster-films > home.spec.js > ...

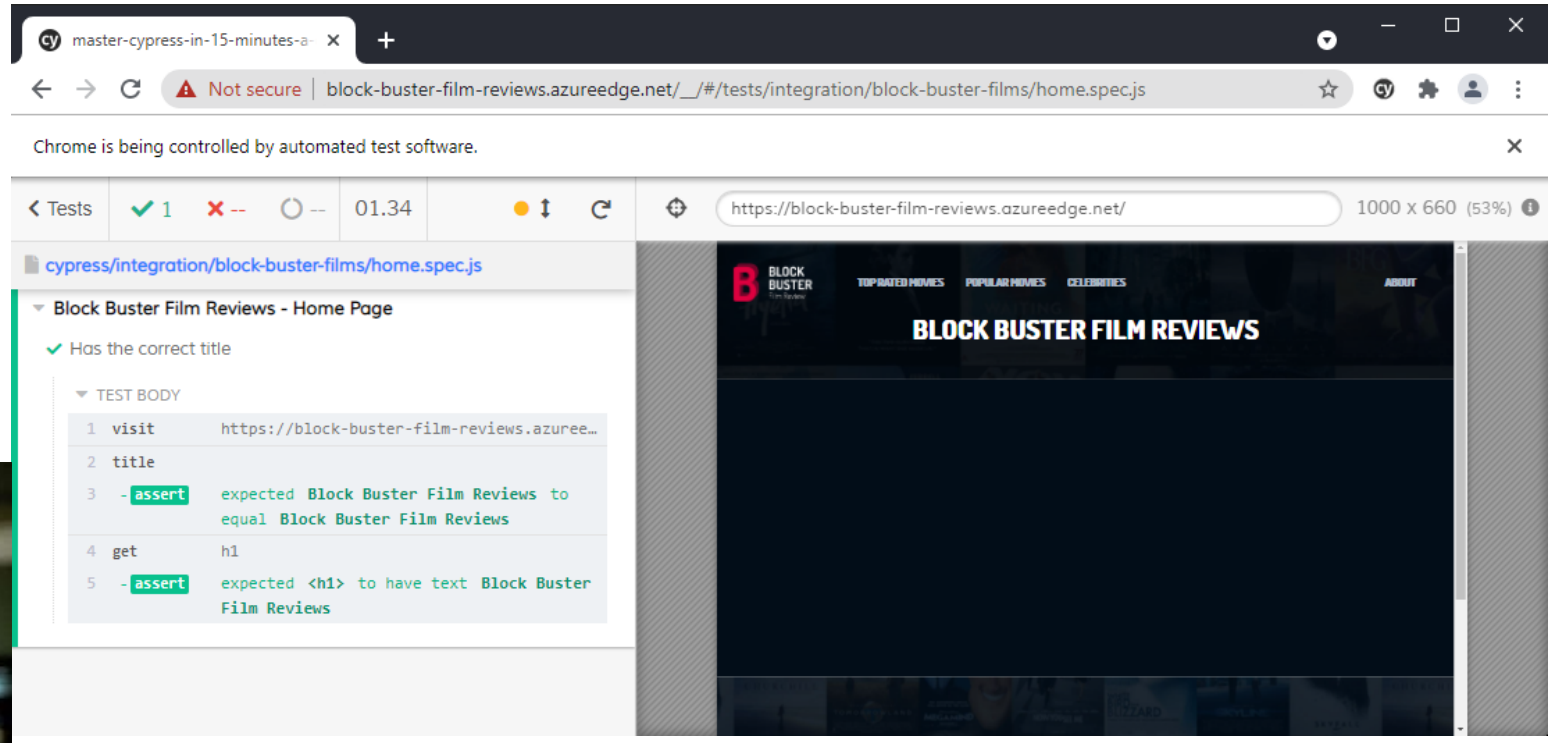
```
1  context('Block Buster Film Reviews - Home Page', () => {
2    it('Has the correct title', () => {
3      cy.visit('https://block-buster-film-reviews.azureedge.net');
4
5      cy.title().should('equal', 'Block Buster Film Reviews');
6      cy.get('h1').should('have.text', 'Block Buster Film Reviews');
7    });
8  });
```

cypress

Cypress test runner



Running the test



cypress

See you in the next video

cypress

Adding a package.json

cypress

Specifying the Cypress version

- Right now we are always using the **latest version of Cypress**
  - Instead of being explicit about what version to use
- Each developer/tester **has to know** to run "npx cypress open"
  - Maybe look for it in the documentation

cypress

Adding a package.json

- When using Node.js the normal way is to **add a package.json**
  - And specify the dependencies with versions in there
  - As well as add important scripts to execute
- **Create a package.json** with the "npm init" command

cypress

Adding dependencies

- **Add dependencies** with "npm install <<package>>"
  - Use the --save-dev option to register a development only package
  - Use "npm install cypress --save-dev"

cypress

Adding scripts

- Add relevant scripts to the “scripts” section
  - “cypress open”
  - “cypress run”
- Execute with “npm run [script name]”

cypress

The package.json

package.json U

package.json > ...

```
{
  "name": "master-cypress-in-15-minutes-a-day",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "cypress run",
    "open": "cypress open"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/mauricedb/master-cypress-in-15-minutes-a-day.git"
  },
  "keywords": [
    "cypress"
  ],
  "author": "Maurice de Beijer",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/mauricedb/master-cypress-in-15-minutes-a-day/issues"
  },
  "homepage": "https://github.com/mauricedb/master-cypress-in-15-minutes-a-day#readme",
  "devDependencies": {
    "cypress": "^7.6.0"
  }
}
```

cypress

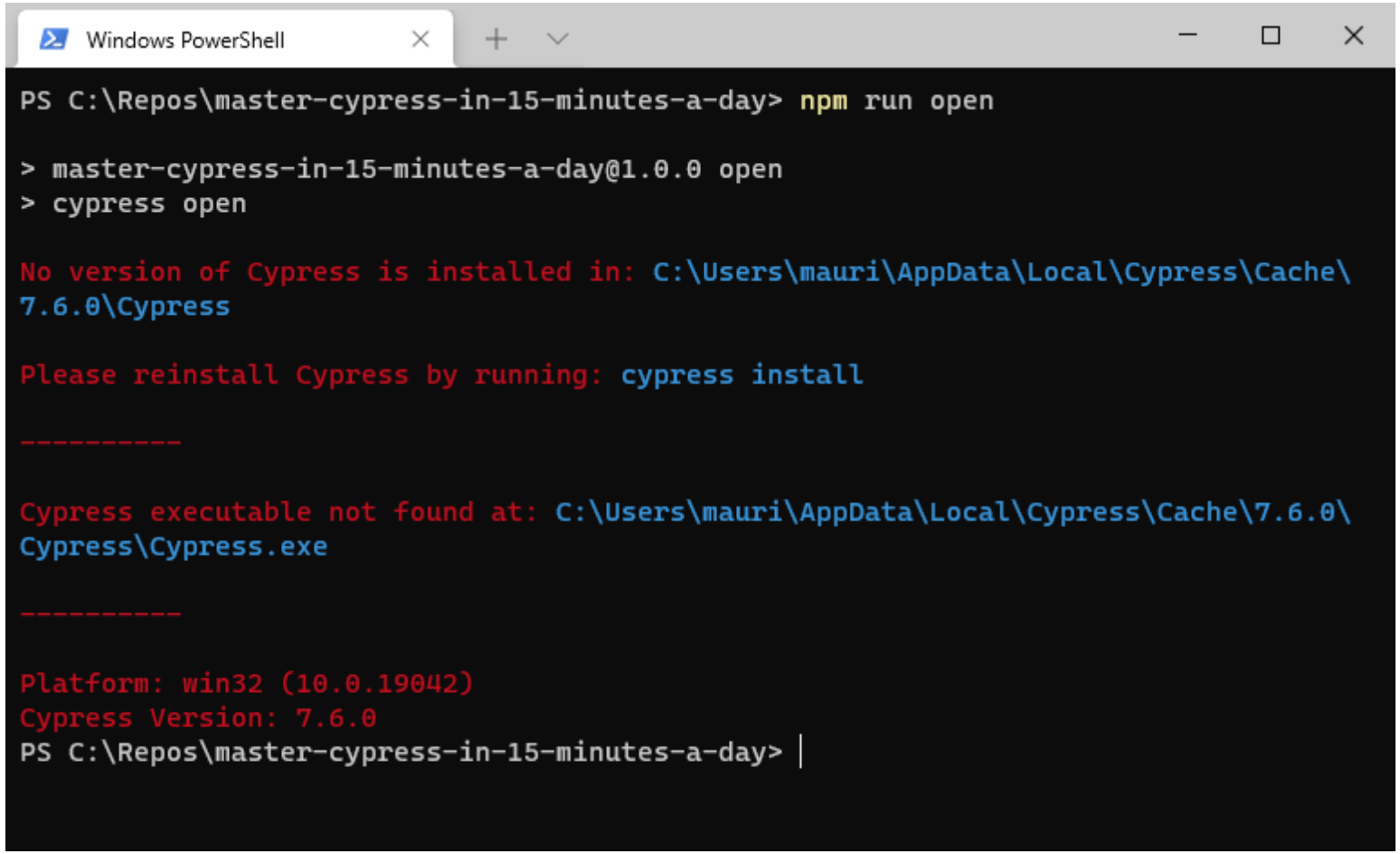
See you in the next video

cypress

Fixing a broken Cypress install

cypress

A broken Cypress Installation



cypress

Cypress NPM Package

- The Cypress NPM package is not complete
  - It only contains part of the required bits
- The rest is installed into an application cache
  - Once for ever version used

cypress

Cypress cache

Binary cache

As of version `3.0`, Cypress downloads the matching Cypress binary to the global system cache, so that the binary can be shared between projects. By default, global cache folders are:

- **MacOS:** `~/Library/Caches/Cypress`
- **Linux:** `~/.cache/Cypress`
- **Windows:** `/AppData/Local/Cypress/Cache`

Cypress cache

- Every version of Cypress is downloaded the first time it's required
  - And reused whenever needed
- Use the "cypress cache" command to manage the Cypress cache
  - npx cypress cache path
  - npx cypress cache list [–size]
  - npx cypress cache prune
  - npx cypress cache clear

cypress

Cypress cache

Windows PowerShell

```
PS C:\Repos\master-cypress-in-15-minutes-a-day> npx cypress cache path
C:\Users\mauri\AppData\Local\Cypress\Cache
PS C:\Repos\master-cypress-in-15-minutes-a-day> npx cypress cache list --size
```

version	last used	size
6.0.1	26 minutes ago	516.1MB
6.5.0	23 minutes ago	609.0MB
6.9.1	20 minutes ago	579.1MB
7.0.1	17 minutes ago	587.4MB
7.3.0	12 minutes ago	574.9MB
7.5.0	14 minutes ago	575.4MB
7.6.0	unknown	444.3MB

```
PS C:\Repos\master-cypress-in-15-minutes-a-day>
```

cypress

See you in the next video

cypress

The most important commands

cypress

cy.visit()

- Visit a URL in the browser
  - Takes the URL to navigate to
  - An optional options object with many settings
- https://docs.cypress.io/api/commands/visit

cypress

cy.get()

- Queries the DOM for one or more elements
  - Uses the jQuery selector engine
- https://docs.cypress.io/api/commands/get

cypress

.should()

- Check if a given condition is as expected
  - Based on the Chai NPM package
  - Often chain after a cy.get()
  - .and() is an alias
- Check for example
  - If an elements is visible and enabled
  - A value is equal to another
  - Many more…
- https://docs.cypress.io/api/commands/should

cypress

Asynchronous

- Commands are asynchronous
  - They not executed immediately but placed in a queue

cypress

Retry behavior

- Commands like get() and should() will retry upon failure
  - By default 4 seconds
- The retry timeout can be configured
  - Either globally or per command

cypress

Chainable

- Most commands return a Chainable object

cypress

Top rated movies Specs

top-rated-movies.spec.js U

cypress > integration > block-buster-films > top-rated-movies.spec.js > ...

```
1  // top-rated-movies.spec.js created with Cypress
2  //
3  // Start writing your Cypress tests below!
4  // If you're unfamiliar with how Cypress works,
5  // check out the link below and learn how to write your first test:
6  // https://on.cypress.io/writing-first-test
7
8  context('Top rated movies', () => {
9    it('Has 20 movies per page', () => {
10     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
11
12     cy.get('.movie-item-style-1').should('have.length', 24)
13   })
14
15   it('Has the correct title', () => {
16     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
17
18     cy.title().should('equal', 'Top rated movies')
19     cy.get('h1')
20       .should('be.visible')
21       .should('have.text', 'Top rated movies')
22   })
23
24   it('The movie should be Dilwale Dulhania Le Jayenge', () => {
25     cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
26
27     cy.get('#movie-19404 > .mv-item-infor > h6 > a')
28       .should('have.text', 'Dilwale Dulhania Le Jayenge')
29   })
```

cypress

See you in the next video

cypress

Adding IntelliSense

cypress

IntelliSense

- **IntelliSense** in VS Code can be added with:
  - `/// <reference types="Cypress" />`
- Uses the **TypeScript type declarations**
  - From the Cypress NPM package

IntelliSense

home.spec.js M

cypress > integration > block-buster-films > home.spec.js > context('Block Buster Film Reviews') callback > it('Has the correct title - Home Page') callback

You, 2 minutes ago | 1 author (You)

```
1  /// <reference types="Cypress" />
2
3  context('Block Buster Film Reviews', () => {
4    it('Has the correct title - Home Page', () => {
5      cy.visit('https://block-buster-film-reviews.azureedge.net')
6
7      cy.title().should('equal', 'Block Buster Film Reviews')
8      cy.get('h1').should('have.text', 'Block Buster Film Reviews')
9    })
10 })
11
```

```
have.text                have.text
have.nested.property
have.returned
have.deep.nested.property
have.length.gte
have.length.lte
have.lengthOf.gte
have.lengthOf.lte
have.always.returned
have.length.greaterThan
have.length
have.length.above
```

cypress

See you in the next video

cypress

Mocha hooks

cypress

Mocha hooks

- Mocha provides **hooks** to run code **before and after tests**
  - The **`before()`** and **`after()`** run once per block
  - The **`beforeEach()`** and **`afterEach()`** run for every test
- Using the **beforeEach()** is very convenient
  - Use `cy.visit()` to load the right page

cypress

Using beforeEach()

top-rated-movies.spec.js M

cypress > integration > block-buster-films > top-rated-movies.spec.js > ...

```
10  context('Top rated movies', () => {
11
12    beforeEach(() => {
13      cy.visit('https://block-buster-film-reviews.azureedge.net/top-rated-movies')
14    })
15
16    it('Has 24 movies per page', () => {
17      cy.get('.movie-item-style-1').should('have.length', 24)
18    })
19
20    it('Has the correct title', () => {
21      cy.title().should('equal', 'Top rated movies')
22      cy.get('h1')
23        .should('be.visible')
24        .and('have.text', 'Top rated movies')
25    })
26
27    it('The movie should be Dilwale Dulhania Le Jayenge', () => {
28      cy.get('#movie-19404 > .mv-item-infor > h6 > a')
29        .should('have.text', 'Dilwale Dulhania Le Jayenge')
30    })
31  })
```

cypress

See you in the next video

cypress

Cypress configuration

cypress

Cypress configuration

- The `cypress.json` is used to store **global configuration values**
  - Base URL
  - Environment variables
  - Retries
  - Timeouts
  - Etc.
- https://docs.cypress.io/guides/references/configuration#cypress-json

IntelliSense

- Add a **`$schema`** key to get **IntelliSense**
  - https://on.cypress.io/cypress.schema.json

cypress

Cypress.config()

- The **`Cypress.config()`** function
  - Retrieve a configuration value
  - Override a configuration value for a single test

cypress

Command Line

- Use the `--config` option to **override config using the CLI**
  - Both with `cypress run` and `cypress open`

cypress

cypress.json

cypress.json M

cypress.json > ...

You, seconds ago | 1 author (You)

```
{
  "$schema": "https://on.cypress.io/cypress.schema.json",
  "baseUrl": "https://block-buster-film-reviews.azureedge.net",
  "defaultCommandTimeout": 10000,
  "ignoreTestFiles": [
    "**/1-getting-started/*",
    "**/2-advanced-examples/*"
  ]
}
```

cypress

See you in the next video

cypress

Selecting DOM elements

cypress

CSS Selectors

- Cypress uses the **jQuery selector engine**
  - Supports the standard CSS queries plus extension
- https://api.jquery.com/category/selectors/

cypress

✕ Anti-Pattern

- Using **highly brittle selectors** that are subject to change
- https://docs.cypress.io/guides/references/best-practices#Selecting-Elements

cypress

✓ Best Practice

- **Isolate selectors** from CSS or JS changes
  - ✗ Avoid: `cy.get('.btn.btn-large').click()`
- Using **data-* attributes**
  - ✓ Use: `cy.get('[data-cy=submit]').click()`

cypress

Recommendations

Selector	Recommended	Notes
`cy.get('button').click()`	⚠ Never	Worst - too generic, no context.
`cy.get('.btn.btn-large').click()`	⚠ Never	Bad. Coupled to styling. Highly subject to change.
`cy.get('#main').click()`	⚠ Sparingly	Better. But still coupled to styling or JS event listeners.
`cy.get('[name=submission]').click()`	⚠ Sparingly	Coupled to the `name` attribute which has HTML semantics.
`cy.contains('Submit').click()`	✓ Depends	Much better. But still coupled to text content that may change.
`cy.get('[data-cy=submit]').click()`	✓ Always	Best. Isolated from all changes.

cypress

.find()

- The `.find()` function **searches descendants** of a previous selection
  - Must be chained of a `cy.get()` or similar function
- https://docs.cypress.io/api/commands/find

cypress

.within()

- **Don't** store the result of a command in a **variable**
  - ❌ You will not get a consistent result
- Use the `.within()` function
  - To **scope queries** to a previous result
- https://docs.cypress.io/api/commands/within

cypress

.eq()

- Use the eq( ) function to get a **specific element** from a **collection**
  - A negative index counts from the end
- https://docs.cypress.io/api/commands/eq

cypress

home.spec.js

home.spec.js M

cypress > integration > block-buster-films > home.spec.js > ...

```
13  context('The left navigation menu', () => {
14    it('The first navigation link points to Top rated movies', () => {
15      cy.get('.menu-left a:visible').first().should('have.text', 'Top rated movies')
16    })
17
18    it('The last  navigation link points to Celebrities', () => {
19      cy.get('.menu-left').find('a:visible').last().should('have.text', 'Celebrities')
20    })
21
22    it('Nested searches', () => {
23      cy.get('.menu-left').within(() => {
24        cy.get('a:visible').first().should('have.text', 'Top rated movies')
25        cy.get('a:visible').last().should('have.text', 'Celebrities')
26
27      })
28    })
29
30    it('The navigation links have the correct text', () => {
31      cy.get('.menu-left a:visible').eq(0).should('have.text', 'Top rated movies')
32      cy.get('.menu-left a:visible').eq(1).should('have.text', 'Popular movies')
33      cy.get('.menu-left a:visible').eq(-1).should('have.text', 'Celebrities')
34    })
35  })
36
37  context.skip('Anti-patterns', () => {
38    it("Don't use variables", () => {
39      // This is an anti-pattern. Don't use!
40      const menuLeft = cy.get('.menu-left')
41      menuLeft.find('a:visible').first().should('have.text', 'Top rated movies')
```

cypress

See you in the next video

cypress

Querying by text

cypress

cy.contains()

- **Searches** for a DOM element **based on the text**
  - Based on a string or regular expression
- A string based search is **case sensitive**
  - Can be controlled with the `matchCase` option
- Can both be use as a **top level or a child command**

cypress

cy.contains()

- Always **returns a single DOM element**
  - The first in case of multiple matches
- Sometimes **returns a higher level DOM element**
  - With buttons, links and labels
  - Can be controlled with a selector

cypress

home.spec.js

JS home.spec.js M

cypress > integration > block-buster-films > JS home.spec.js > ...

```
38      it('Can navigate to Top rated movies', () => {
39        cy.contains('Top rated movies').click()
40        cy.title().should('equal', 'Top rated movies')
41      })
42
43      it('Can navigate to Popular movies', () => {
44        cy.contains(/^popular movies$/i).click()
45        cy.title().should('equal', 'Popular movies')
46      })
47
48      it('Can navigate to Celebrities', () => {
49        cy.get('.menu-left').contains('Celebrities').click()
50        cy.title().should('equal', 'Celebrities')
51      })
52    })
```

cypress

See you in the next video

cypress

Aliases

cypress

Aliases

- DOM queries can be **aliased for later reuse**
  - Alias a query:
    `cy.get('.menu-left a:visible').as('nav-links')`
  - Use the alias:
    `cy.get('@nav-links').eq(0).click()`
- Aliases can also be use with other concepts
  - AJAX requests, spies etc.
  - More about that in another video

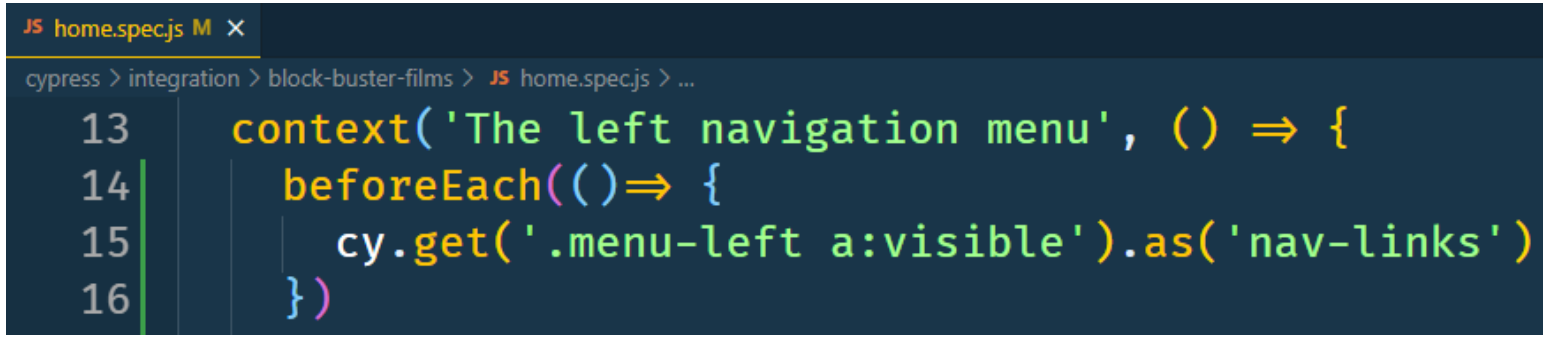
cypress

Aliases and Retries

- The query is executed when first defined
  - And the result reused later
  - Querying will happen again when assertions fail

cypress

Register an alias with .as()



cypress

Using an alias with .get()

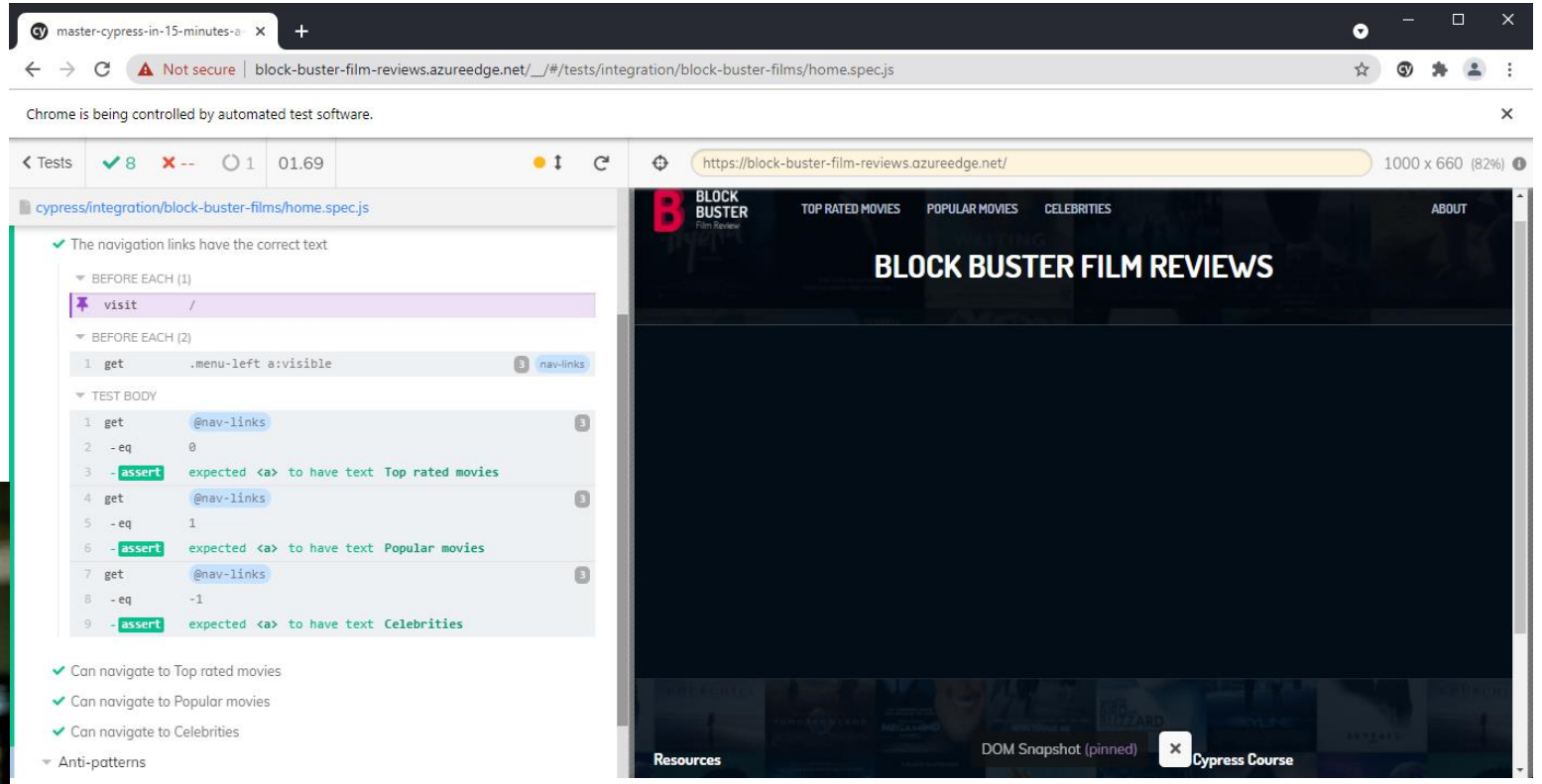
home.spec.js M

cypress > integration > block-buster-films > home.spec.js > context('Block Buster Film Reviews') callback > context('The left navigation menu') callback

```
37    it('The navigation links have the correct text', () => {
38      cy.get('@nav-links').eq(0).should('have.text', 'Top rated movies')
39      cy.get('@nav-links').eq(1).should('have.text', 'Popular movies')
40      cy.get('@nav-links').eq(-1).should('have.text', 'Celebrities')
41    })
```

cypress

The result



cypress

See you in the next video

cypress

Interacting with Elements

cypress

Common actions

- **Input actions:**
  - Typing into inputs
  - Checking checkboxes
  - Selecting options
  - Etc.
- **Mouse actions:**
  - Clicking
  - Scrolling
  - Etc.

cypress

See you in the next video

cypress

Clicking DOM elements

cypress

cy.click()

- With `cy.click()` you can **click on a DOM element**
  - All events are simulated by Cypress
- The element will **scroll into view**
- You can specify the **click position**
  - With coordinates or a position string
- Cypress checks if the **element is enabled and visible**
  - Use `{ force: true }` to override when needed
- https://docs.cypress.io/api/commands/click

cypress

top-rated-movies.spec.js

JS top-rated-movies.spec.js M ×

cypress > integration > block-buster-films > JS top-rated-movies.spec.js > ...

```
31    it('Open Pulp Fiction', () => {
32      cy.get('#movie-680 .hvr-inner > a')
33        .click('bottomRight', { force: true })
34    })
35  })
```

cypress

See you in the next video

cypress

Typing text

cypress

cy.type()

- With **cy.type()** you can simulate typing
  - Works with <input> and <textarea>
- Special keys can be entered using **{} expressions**
  - Like {enter} or {backspace}
  - Or modifier keys like {alt} or {ctrl}
- There is a **10 ms delay** between key events
  - Makes the event flow more realistic
- https://docs.cypress.io/api/commands/type

cypress

todo-list.spec.js

todo-list.spec.js U

cypress > integration > demos > todo-list.spec.js > ...

```
3  context('To Do List', () => {
4    beforeEach(() => {
5      cy.visit('/')
6      cy.get('[data-testid=new-todo]').as('new-todo')
7      cy.get('[data-testid=btn-add-todo]').as('btn-add-todo')
8    })
9
10   it('Can add a new to-do with button', () => {
11     cy.get('@new-todo').type('A new to-do with button')
12     cy.get('@btn-add-todo').click()
13
14     cy.contains('li', 'A new to-do with button').should('be.visible')
15   })
```

cypress

todo-list.spec.js

todo-list.spec.js U

cypress > integration > demos > todo-list.spec.js > ...

```
17  it('Can add a new to-do with enter', () => {
18    cy.get('@new-todo').type('A new to-do with enter{enter}')
19
20    cy.contains('li', 'A new to-do with enter').should('be.visible')
21  })
22
23  it('Can add a new to-do in steps', () => {
24    cy.get('@new-todo')
25      .type('A ')
26      .type('to-do ')
27      .type('in ')
28      .type('steps')
29      cy.get('@btn-add-todo').click()
30
31    cy.contains('li', 'A to-do in steps').should('be.visible')
32  })
```

cypress

settings.spec.js

settings.spec.js U

cypress > integration > demos > settings.spec.js > ...

```
3  context('Settings', () => {
4    beforeEach(() => {
5      cy.visit('/')
6      cy.get('#view #settings').click()
7    })
8
9    it('Change the name to Jack Sparrow', () => {
10     cy.get('#firstName').clear().type('Jack')
11     cy.get('#lastName').type('{selectAll}Sparrow')
12     cy.get('#email').type('captain@black-pearl.ship')
13
14     cy.get('#result')
15       .should('contain.value', '"firstName": "Jack",')
16       .should('contain.value', '"lastName": "Sparrow",')
17       .should('contain.value', '"email": "captain@black-pearl.ship",')
18   })
19 })
```

cypress

todo-list.spec.js

todo-list.spec.js U

cypress > integration > demos > todo-list.spec.js > ...

```
34  it('Can add {abc}', () => {
35    cy.get('@new-todo').type('Do {abc}', {
36      parseSpecialCharSequences: false
37    })
38    cy.get('@btn-add-todo').click()
39
40    cy.get('@new-todo').type('Do {{}xyz}{enter}')
41
42    cy.contains('li', 'Do {abc}').should('be.visible')
43  })
```

cypress

See you in the next video

cypress

Radio buttons & checkboxes

cypress

cy.check()

- With **cy.check()** you can check an item
  - Works with <input type="radio"> and <input type="checkbox">
- https://docs.cypress.io/api/commands/check

cypress

settings.spec.js

settings.spec.js M

cypress > integration > demos > settings.spec.js > ...

```
20    it('Change the gender to female', () => {
21      cy.get('#result').should('contain.value', '"gender": ""')
22
23      cy.get('#female').check()
24
25      cy.get('#result').should('contain.value', '"gender": "female"')
26    })
27
28    it('Select the newsletter', () => {
29      cy.get('#result').should('contain.value', '"newsletter": false')
30
31      cy.get('#newsletter').check()
32
33      cy.get('#result').should('contain.value', '"newsletter": true')
34    })
35  })
```

cypress

See you in the next video

cypress

Select elements

cypress

cy.select()

- With **cy.select()** you can select an option
  - Works with <select>
- Use either the **visible text or the value**
  - Use an array with multiselecting
- https://docs.cypress.io/api/commands/select

cypress

settings.spec.js

settings.spec.js M

cypress > integration > demos > settings.spec.js > ...

```
38    it('Change the country to NL', () => {
39      cy.get('#result').should('contain.value', '"country": "US"')
40
41      cy.get('#country').select('Netherlands')
42
43      cy.get('#result').should('contain.value', '"country": "NL"')
44    })
45  })
```

cypress

See you in the next video

cypress

Manipulating the value

cypress

Manipulating Values

- Some input elements are **harder to manipulate** like an end user
  - For example:
    - `<input type="date"/>`
    - `<input type="range"/>`
- **Manipulate the value directly** using:
  -
```
cy.get("input[type=range]")
  .invoke("val", 25).trigger("change")
```
- Note: Sometimes you need to trigger the **input event**
  - Instead of **change event**

settings.spec.js

settings.spec.js M

cypress > integration > demos > settings.spec.js > ...

```
46    it('Change the happiness to 95%', () => {
47      cy.get('#result').should('contain.value', '"happiness": 80')
48
49      cy.get('#happiness').invoke('val', 95).trigger('input')
50
51      cy.get('#result').should('contain.value', '"happiness": 95')
52    })
53  })
```

cypress

See you in the next video

cypress

Validating the value

cypress

Validating the value

- **Validating inputs** is normally done using the `.should()` function
- Many **validation options** you can use
  - Have a specific value
  - Is checked or not checked
  - Is empty
  - Etc.
- https://docs.cypress.io/api/commands/should

cypress

settings.spec.js

settings.spec.js M

cypress > integration > demos > settings.spec.js > ...

```
9   it('Change the name to Jack Sparrow', () => {
10    cy.get('#firstName')
11      .clear()
12      .type('Jack')
13      .should('have.value', 'Jack')
14    cy.get('#lastName')
15      .type('{selectAll}Sparrow')
16      .should('have.value', 'Sparrow')
17    cy.get('#email')
18      .type('captain@black-pearl.ship')
19      .should('have.value', 'captain@black-pearl.ship')
20  })
21
22  it('Change the gender to female', () => {
23    cy.get('#unknown')
24      .should('be.checked')
25
26      cy.get('#female')
27      .check()
28      .should('be.checked')
29
30    cy.get('#unknown')
31      .should('not.be.checked')
32  })
```

cypress

See you in the next video

cypress

Network Requests & Cypress

cypress

Network Requests

- **Making network requests** with `cy.request()`
  - Dealing with cookies
  - Handling error responses
- **Intercepting network requests** with `cy.intercept()`
  - Waiting for request to have finished
  - Inspecting the request or response
  - Faking the response message

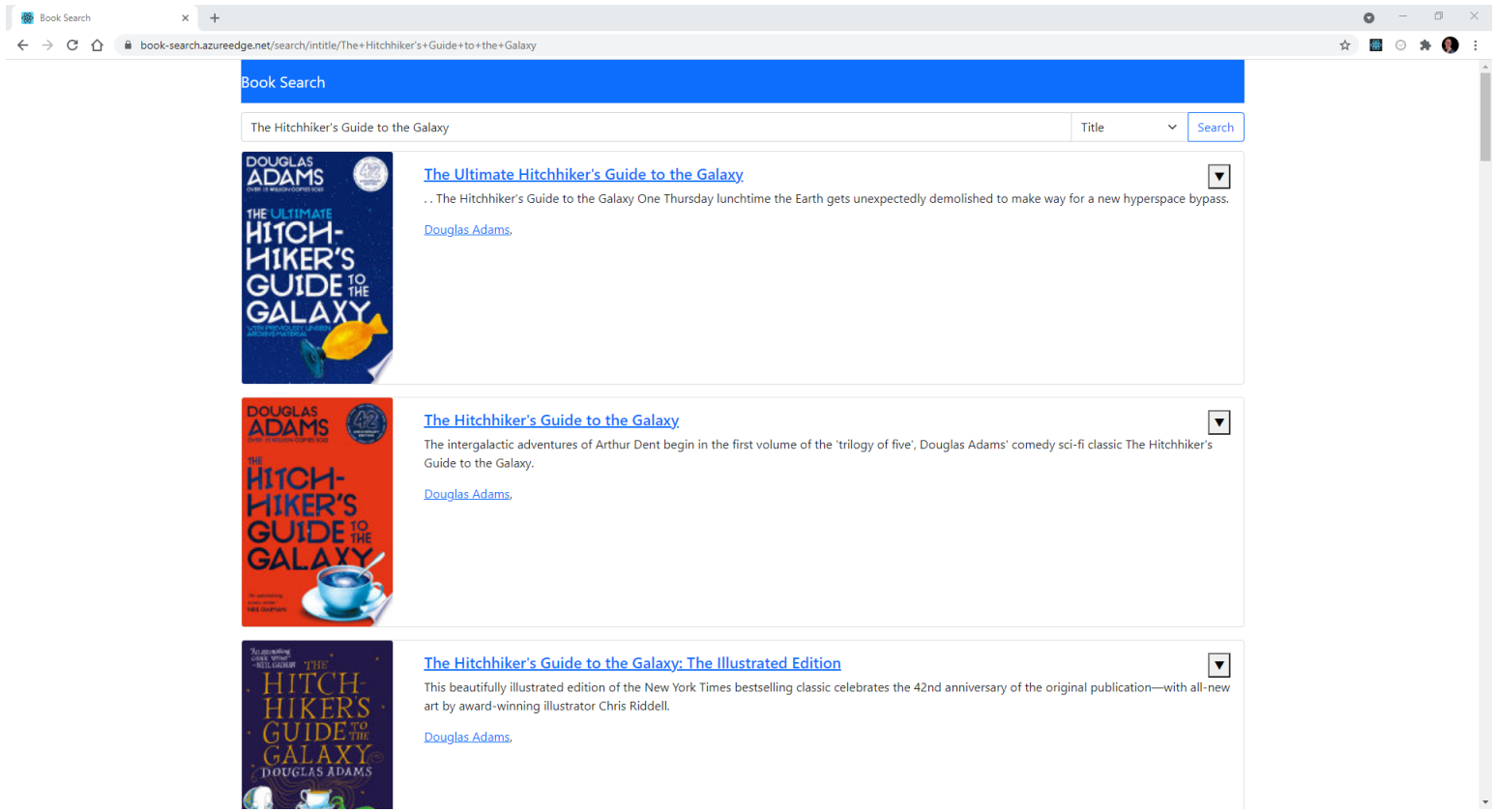
cypress

See you in the next video

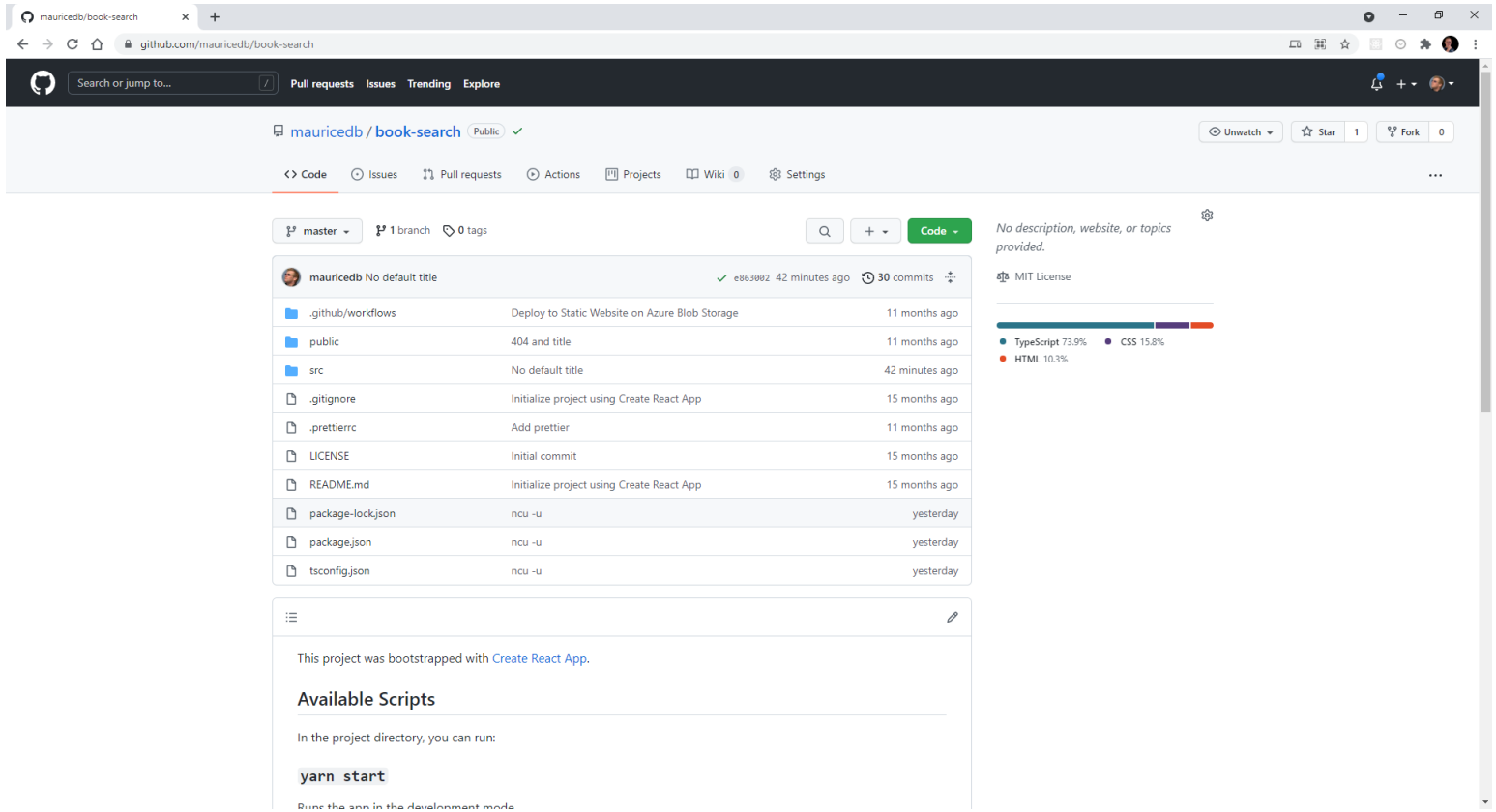
cypress

Book Search

Book Search


Book Search
book-search.azureedge.net/search/intitle/The+Hitchhiker's+Guide+to+the+Galaxy
Book Search
The Hitchhiker's Guide to the Galaxy
Title
Search
The Ultimate Hitchhiker's Guide to the Galaxy
. . The Hitchhiker's Guide to the Galaxy One Thursday lunchtime the Earth gets unexpectedly demolished to make way for a new hyperspace bypass.
Douglas Adams,
The Hitchhiker's Guide to the Galaxy
The intergalactic adventures of Arthur Dent begin in the first volume of the 'trilogy of five', Douglas Adams' comedy sci-fi classic The Hitchhiker's Guide to the Galaxy.
Douglas Adams,
The Hitchhiker's Guide to the Galaxy: The Illustrated Edition
This beautifully illustrated edition of the New York Times bestselling classic celebrates the 42nd anniversary of the original publication—with all-new art by award-winning illustrator Chris Riddell.
Douglas Adams,

GitHub


mauricedb/book-search
github.com/mauricedb/book-search
Search or jump to...
Pull requests
Issues
Trending
Explore
mauricedb / book-search
Public
Unwatch
Star 1
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki 0
Settings
master
1 branch
0 tags
Code
No description, website, or topics provided.
MIT License
mauricedb No default title
e863002 42 minutes ago
30 commits
.github/workflows Deploy to Static Website on Azure Blob Storage 11 months ago
public 404 and title 11 months ago
src No default title 42 minutes ago
.gitignore Initialize project using Create React App 15 months ago
.prettierrc Add prettier 11 months ago
LICENSE Initial commit 15 months ago
README.md Initialize project using Create React App 15 months ago
package-lock.json ncu -u yesterday
package.json ncu -u yesterday
tsconfig.json ncu -u yesterday
TypeScript 73.9%
CSS 15.8%
HTML 10.3%
This project was bootstrapped with Create React App.
Available Scripts
In the project directory, you can run:
yarn start

search.spec.js

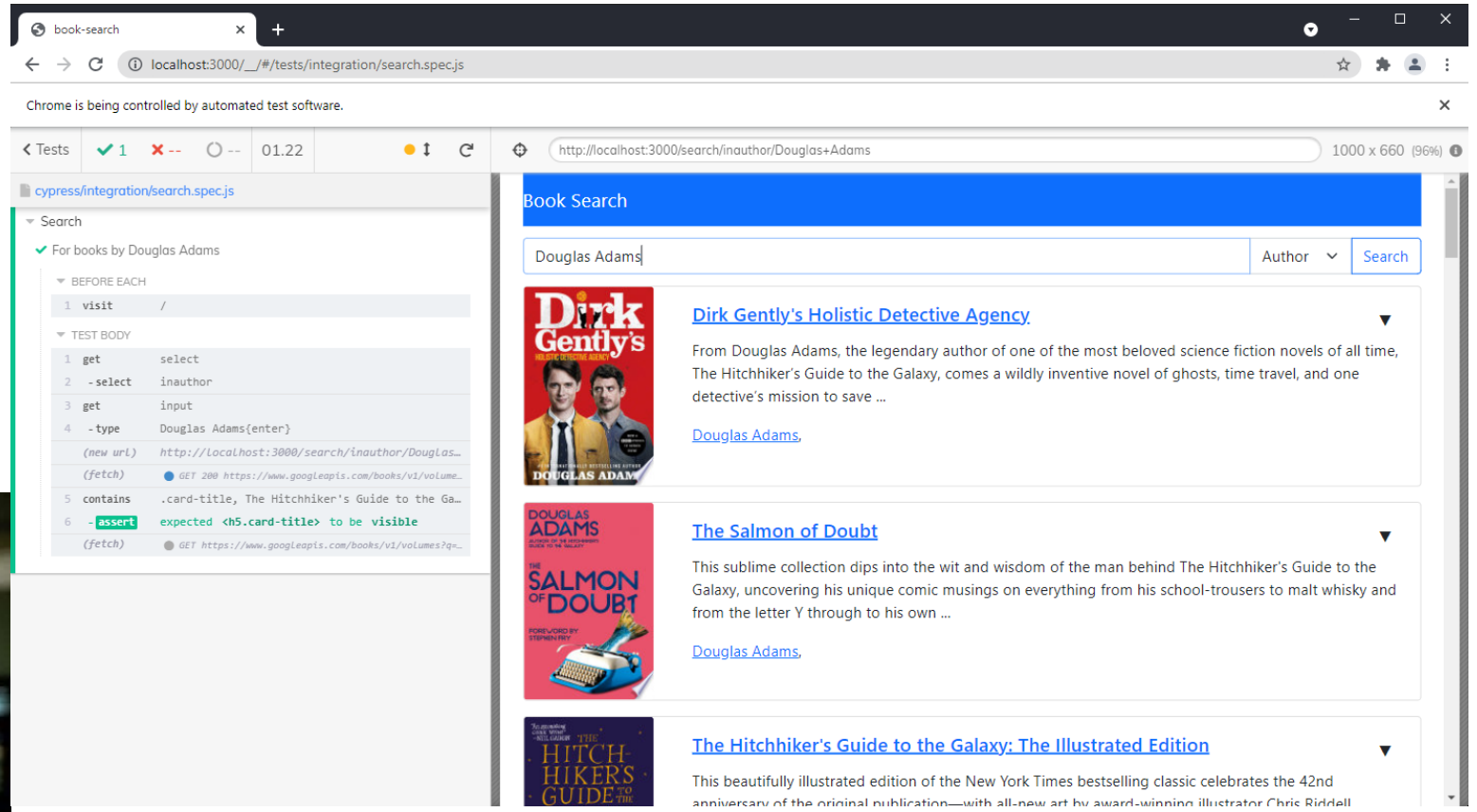
search.spec.js 4, U

cypress > integration > search.spec.js > ...

```
1  /// <reference types="cypress" />
2
3  describe('Search', () => {
4    beforeEach(() => {
5      cy.visit('/');
6    });
7
8    it('For books by Douglas Adams', () => {
9      cy.get('select').select('inauthor');
10     cy.get('input').type('Douglas Adams{enter}');
11
12     cy.contains('.card-title', "The Hitchhiker's Guide to the Galaxy").should(
13       'be.visible'
14     );
15   });
16 });
```

cypress

Cypress open



cypress

See you in the next video

cypress

Intercepting Requests

cypress

cy.intercept()

- With `cy.intercept()` you can **spy on network requests**
  - Both `XMLHttpRequest` and `fetch` requests
- **Enables waiting for** network requests to **finish**
  - Before proceeding with a test
- **Intercept AJAX requests**
  - Assert the request message
  - Fake response messages
- **Intercept AJAX responses**
  - Delay a response
  - Use the response in the test body
- https://docs.cypress.io/api/commands/intercept

cypress

cy.intercept()

search.spec.js M

cypress > integration > search.spec.js > ...

```
8   it('intercept with a StringMatcher', () => {
9     cy.intercept('/books/v1/volumes?*').as('book-search');
10
11    cy.get('select').select('inauthor');
12    cy.get('input').type('Douglas Adams{enter}');
13
14    cy.contains('.card-title', "The Hitchhiker's Guide to the Galaxy").should(
15      'be.visible'
16    );
17  });
18
19  it('intercept with a RouteMatcher', () => {
20    cy.intercept({
21      hostname: 'www.googleapis.com',
22      method: 'get',
23      pathname: '/books/v1/volumes',
24    }).as('book-search');
25
26    cy.get('select').select('inauthor');
27    cy.get('input').type('Douglas Adams{enter}');
28
29    cy.contains('.card-title', "The Hitchhiker's Guide to the Galaxy").should(
30      'be.visible'
31    );
32  });
```

cypress

See you in the next video

cypress

Cypress.minimatch()

Cypress.minimatch()

- Cypress uses the minimatch NPM package to match URL's
- Can be used to debug URL patterns
  - Available using `Cypress.minimatch()`
- https://docs.cypress.io/api/utilities/minimatch
- https://github.com/isaacs/minimatch#minimatch

cypress

Cypress.minimatch()

cypress/integration/search.spec.js

```
3  get          input
4  -type        Douglas Adams{enter}
   (new url)    http://localhost:3000/search/inauthor/Douglas+Adams
   (fetch)      GET 200 https://www.googleapis.com/books/v1/volumes?q=inautho…  book-search
5  contains     .card-title, The Hitchhiker's Guide to the Galaxy
6  -assert      expected <h5.card-title> to be visible
```

DOM

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse

top ▾ | Filter

```
Console was cleared
Event:                          request
Resource type:                  fetch
Method:                         GET
Url:                            https://www.googleapis.com/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks
Request went to origin?:        yes
Matched `cy.intercept()`:       ▶ {RouteMatcher: {…}, RouteHandler Type: 'Spy', RouteHandler: undefined, Request: {…}, Response: {…}, …}
Request headers:
▶ {sec-ch-ua: '"Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"', Referer: 'http://localhost:3000/', sec-ch-ua-mobile: '?0', User-Agent:
  sec-ch-ua-platform: '"Windows"'}
Response status code:           200
Response headers:               ▶ {date: 'Fri, 24 Sep 2021 07:08:02 GMT', content-encoding: 'gzip', x-content-type-options: 'nosniff', server: 'ESF', Vary: 'Or
Response body:                  ▶ {kind: 'books#volumes', totalItems: 300, items: Array(25)}
> Cypress.minimatch('/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks', '/books/v1/volumes?*', { matchBase: true })
< true
> Cypress.minimatch('/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks', '/books/v1/volumes', { matchBase: true })
< false
> Cypress.minimatch('/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks', '/volumes?*', { matchBase: true })
< false
> Cypress.minimatch('/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks', '*/volumes?*', { matchBase: true })
< false
> Cypress.minimatch('/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&filter=ebooks', '**/volumes?*', { matchBase: true })
< true
```

cypress

See you in the next video

cypress

cy.route() & cy.server()

cypress

cy.route()
cy.server()

- Both `cy.route()` and `cy.server()` have been **deprecated**
  - Use `cy.intercept()` instead!

cypress

See you in the next video

cypress

Waiting for a response

cypress

cy.wait()

- Waits for a network request to complete
  - It can wait on multiple requests
- Fails the test if the requests doesn't complete in time
- https://docs.cypress.io/api/commands/wait

cypress

cy.wait()

search.spec.js M

cypress > integration > search.spec.js > ...

```
41  it('waiting for a fetch()', () => {
42    cy.intercept('/books/v1/volumes?*').as('book-search');
43
44    cy.get('select').select('inauthor');
45    cy.get('input').type('Douglas Adams{enter}');
46
47    cy.wait('@book-search');
48
49    cy.contains('.card-title', "The Hitchhiker's Guide to the Galaxy").should(
50      'be.visible'
51    );
52  });
```

cypress

See you in the next video

cypress

Faking AJAX responses

cypress

Faking AJAX responses

- Use a `staticResponse` to return a predetermined response
  - Preventing the original AJAX request
- Set the `statusCode`, `body` and/or `headers` options as needed
- The `fixture` option loads a file from the `cypress/fixtures` folder
  - The fixture location is configurable
- https://docs.cypress.io/api/commands/intercept#staticResponse-lt-code-gtStaticResponselt-code-gt

cypress

Faking AJAX responses

Pro's

- Faster
- No dependencies on the backend services
- No need to reset the DB between tests
- No issues with external API's

Con's

- Doesn't test the application as it will be used
- The API response might be updated

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
54    it('respond with fake data', () => {
55      cy.intercept('/books/v1/volumes?*', { fixture: 'douglas-adams.json' }).as(
56        'book-search'
57      );
58
59      cy.get('select').select('inauthor');
60      cy.get('input').type('Douglas Adams{enter}');
61
62      cy.contains('.card-title', 'Book two').should('be.visible');
63    });
```

cypress

See you in the next video

cypress

Intercepting the response

cypress

Intercepting the response

- Using `cy.wait()` with an AJAX request yields the `response` object
  - With a JSON body object if appropriate
- Use the response data to test the UI elements
- https://docs.cypress.io/api/commands/wait#Yields

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
54    it('intercept and use the response', () => {
55      cy.intercept('/books/v1/volumes?*').as('book-search');
56
57      cy.get('select').select('inauthor');
58      cy.get('input').type('Douglas Adams{enter}');
59
60      cy.wait('@book-search').then((xhr) => {
61        const books = xhr.response.body.items;
62
63        books.forEach((book, index) => {
64          cy.get('.card')
65            .eq(index)
66            .within(() => {
67              cy.get('.card-title').should('have.text', book.volumeInfo.title);
68              cy.get('.card-author').should(
69                'have.text',
70                book.volumeInfo.authors.join(', ')
71              );
72            });
73        });
74      });
75    });
76  });
```

cypress

See you in the next video

cypress

Making network requests

cypress

cy.request()

- Using `cy.request()` you can make network requests from a test
  - Useful for testing your API's
  - Submitting URL encoded forms
  - Calling an API to seed test data
  - Etc…
- https://docs.cypress.io/api/commands/request

cypress

Login example

request.spec.js M

cypress > integration > request.spec.js > ...

```
6   it.skip('to login and retreive the token', () => {
7     cy.request('POST', '/login', {
8       username: 'maurice.de.beijer',
9       password: 'p@ssw0rd',
10    })
11      .its('body.token')
12      .then((token) => {
13        console.log(`Token: ${token}`);
14      });
15  });
```

cypress

request.spec.js

request.spec.js M

cypress > integration > request.spec.js > ...

```
17  it('to search for Douglas Adams', () => {
18    cy.request(
19      'https://www.googleapis.com/books/v1/volumes?q=inauthor:Douglas+Adams&maxR
20    ).as('response');
21
22    cy.get('@response').its('body.kind').should('equal', 'books#volumes');
23    cy.get('@response').its('body.totalItems').should('be.greaterThan', 350);
24    cy.get('@response').its('body.items').should('have.length', 25);
25    cy.get('@response').its('body.items.0.volumeInfo.title').should('exist');
26    cy.get('@response').its('body.items.0.volumeInfo.authors').should('exist');
27  });
```

cypress

See you in the next video

cypress

Continuous Integration

cypress

Continuous Integration

- Run your Cypress tests as part of your CI process
  - Make sure all tests pass before merging a pull request
- Tests need to run against the local code in the PR
  - Run your application with a development server
- The steps can vary between different environments
  - But the same principal applies everywhere

cypress

See you in the next video

cypress

CI using GitHub Actions

cypress

GitHub Actions

- Run Cypress as part of a **GitHub CI/CD pipeline**
  - Only deploy the application if the tests pass
- Run the development server on the **GitHub infrastructure**
  - Run the Cypress tests against this GitHub hosted server
- The **cypress-io/github-action** often makes this easy
  - Supports most of the common use cases
- https://docs.cypress.io/guides/continuous-integration/github-actions

cypress

cypress-io.yml

cypress-io.yml U

.github > workflows > cypress-io.yml

```yaml
name: Cypress CI Tests

on:
  # Manual button click from the GitHub UI
  workflow_dispatch:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  start:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Code
        uses: actions/checkout@v2

      - name: Run Cypress E2E Tests
        uses: cypress-io/github-action@v2
        with:
          start: npm start
          wait-on: 'http://localhost:3000'

      - name: Upload Cypress Screenshots
        uses: actions/upload-artifact@v2
        if: failure()
        with:
          name: Cypress screenshots
          path: cypress/screenshots
```

cypress

See you in the next video

cypress

Cypress Continuous Integration Steps

cypress

Cypress CI Steps

- The **basic steps** for a Cypress continuous integration process are:
  1. Get the latest version of the application
  2. Run `npm install`
  3. Start the web server
  4. Execute a `cypress run`
  5. `Report the result`

cypress

Booting the web server

- **Booting the web server** can be tricky
  - It should not block the process
  - The Cypress tests should only start when the app is available
- The start-server-and-test NPM package is very useful

cypress

cypress-io.yml

cypress-io.yml M

.github > workflows > cypress-io.yml

```yaml
steps:
  - name: Checkout Code
    uses: actions/checkout@v2

  - name: NPM Install
    run: npm ci

  - name: Install Start Server and Test
    run: npm install start-server-and-test --no-save

  - name: Run Application and Cypress E2E Tests
    run: npx start-server-and-test start http://localhost:3000 cypress:run

  - name: Upload Cypress Screenshots
    uses: actions/upload-artifact@v2
    if: failure()
    with:
      name: Cypress screenshots
      path: cypress/screenshots

  - name: Upload Cypress Videos
    uses: actions/upload-artifact@v2
    if: failure()
    with:
      name: Cypress videos
      path: cypress/videos
```

cypress

See you in the next video

cypress

Using a Docker container

cypress

Docker containers

- Using Docker containers can help create a **consistent environment**
  - The same runtime locally and on the CI server
- Required when doing **visual regression testing**
  - Often a good approach
- https://hub.docker.com/r/cypress/included

cypress

cypress-io.yml

cypress-io.yml

.github > workflows > ! cypress-io.yml

```yaml
jobs:
  start:
    runs-on: ubuntu-latest
    container:
      image: cypress/included:8.5.0
      options: --ipc=host

    steps:
      - name: Checkout Code
        uses: actions/checkout@v2
```

cypress

See you in the next video

cypress

Flaky tests

cypress

Flaky tests

- Flaky tests are tests that **sometimes fail** and sometimes pass without any change in code
- Cypress can **automatically retry failing tests** to detect flakiness
  - Helps prevent a complete CI build from failing
  - The underlying flaky test should still be fixed

cypress

cypress.json

cypress.json M

cypress.json > ...

You, seconds ago | 1 author (You)

```
{
  "$schema": "https://on.cypress.io/cypress.schema.json",
  "baseUrl": "http://localhost:3000",
  "retries": {
    "runMode": 3,
    "openMode": 1
  },
  "ignoreTestFiles": ["**/1-getting-started/*", "**/2-advanced-examples/*"]
}
```

cypress

See you in the next video

cypress

The Cypress Dashboard

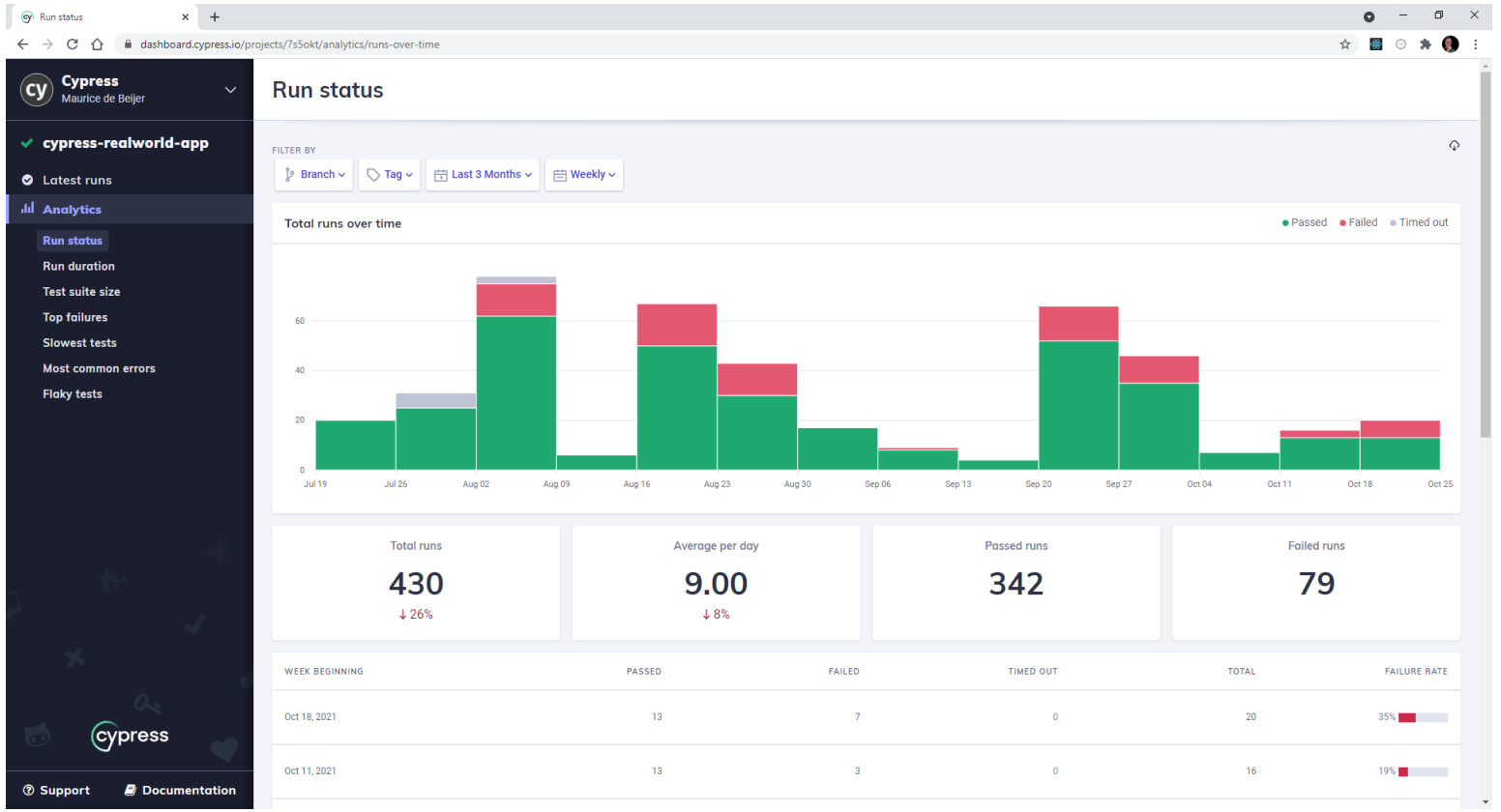
cypress

Cypress Dashboard

- Makes it easy to **see what is happening** on the CI server
- **Saves a video** of the test running
- **Not required**, totally optional
  - This is a paid for option
  - With a limited free tier to get started

cypress

Cypress Dashboard



cypress-io.yml

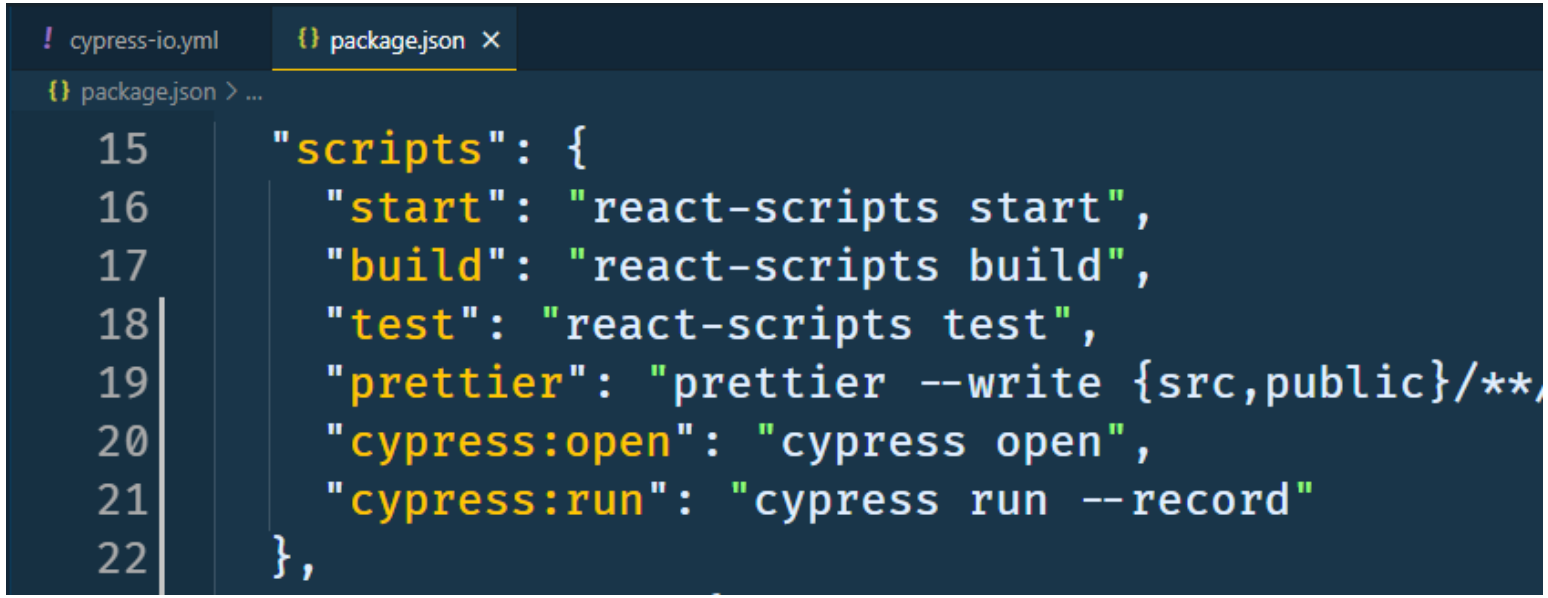
cypress-io.yml × | package.json

.github > workflows > cypress-io.yml

```yaml
jobs:
  start:
    runs-on: ubuntu-latest
    container:
      image: cypress/included:8.6.0
      options: --ipc=host
    env:
      CYPRESS_PROJECT_ID: ${{ secrets.CYPRESS_PROJECT_ID }}
      CYPRESS_RECORD_KEY: ${{ secrets.CYPRESS_RECORD_KEY }}

    steps:
      - name: Checkout Code
        uses: actions/checkout@v2

      - name: NPM Install
        run: npm ci

      - name: Install Start Server and Test
        run: npm install start-server-and-test --no-save

      - name: Run Application and Cypress E2E Tests
        run: npx start-server-and-test start http://localhost:3000 cypress:run
```

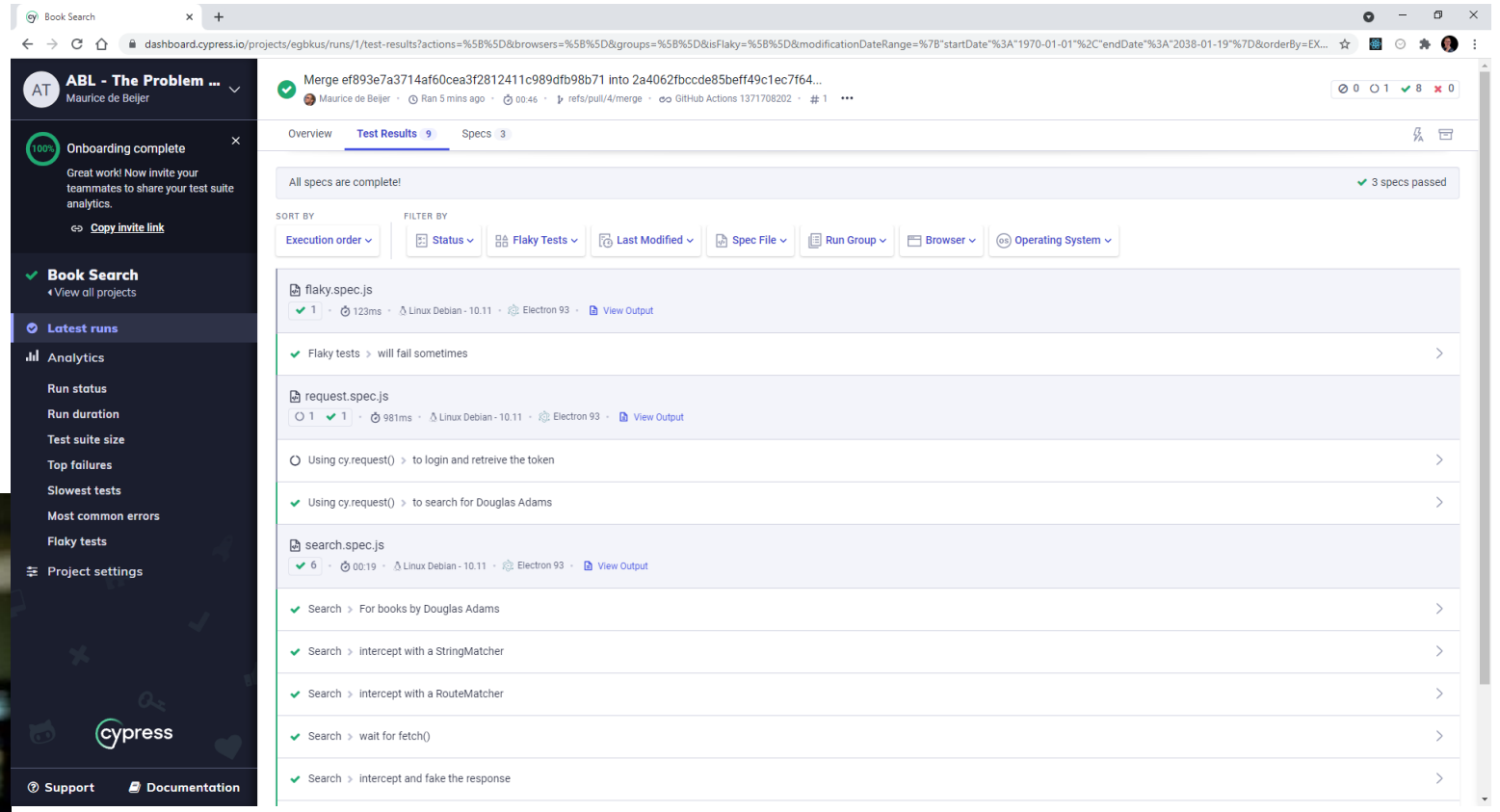
cypress

package.json



cypress

The dashboard



cypress

See you in the next video

cypress

Local GitHub Actions

cypress

Local GitHub Actions

- Using Act you can run GitHub Actions on a local machine
  - Much easier to test and debug actions
- Runs in a local Docker container
- Cypress tests needs to run in a cypress/included image
  - To prevent: Your system is missing the dependency: Xvfb
- Use a .secrets and/or a .env file to configure the environment
  - Or skip steps if needed

cypress

Running Act

Windows PowerShell

```
PS C:\Repos\cypress-tutorial-demos\book-search> act
[Cypress CI Tests/start] 🚀  Start image=cypress/included:9.1.1
[Cypress CI Tests/start]   🐳  docker run image=cypress/included:9.1.1 platform= entrypoint=["/usr/bin/tail" "-f" "/dev/null"] cmd=[]
[Cypress CI Tests/start]   🐳  docker exec cmd=[mkdir -m 0777 -p /var/run/act] user=root
[Cypress CI Tests/start]   🐳  docker cp src=C:\Repos\cypress-tutorial-demos\book-search\. dst=\mnt\c\Repos\cypress-tutorial-demos\book-search
[Cypress CI Tests/start]   🐳  docker exec cmd=[mkdir -p \mnt\c\Repos\cypress-tutorial-demos\book-search] user=
[Cypress CI Tests/start] *  Run Checkout Code
[Cypress CI Tests/start]   ✅  Success - Checkout Code
[Cypress CI Tests/start] *  Run Run Cypress E2E Tests
time="2021-12-15T14:37:23+01:00" level=info msg="  ☁  git clone 'https://github.com/cypress-io/github-action' # ref=v2"
[Cypress CI Tests/start]   🐳  docker cp src=C:\Users\mauri\.cache\act/cypress-io-github-action@v2/ dst=/var/run/act/actions/cypress-io-github-action@v2/
[Cypress CI Tests/start]   🐳  docker exec cmd=[mkdir -p /var/run/act/actions/cypress-io-github-action@v2/] user=
[Cypress CI Tests/start]   🐳  docker exec cmd=[node /var/run/act/actions/cypress-io-github-action@v2/dist/index.js] user=
[Cypress CI Tests/start]   💬  ::debug::Resolved Keys:
[Cypress CI Tests/start]   💬  ::debug::["npm-linux-x64-3de529f50eafb2251381de866ff8199033b27c3ddba611a254aeb57060252ff2dbc7ab3bd26577daff1f874d1a4bee3dd19bae7a35b55bfa252b62d87af365bc","npm-linux-x64-3de529f5
0eafb2251381de866ff8199033b27c3ddba611a254aeb57060252ff2dbc7ab3bd26577daff1f874d1a4bee3dd19bae7a35b55bfa252b62d87af365bc"]
[Cypress CI Tests/start]   💬  ::debug::Checking zstd --version
[Cypress CI Tests/start]   💬  ::debug::Resolved Keys:
[Cypress CI Tests/start]   💬  ::debug::["cypress-linux-x64-3de529f50eafb2251381de866ff8199033b27c3ddba611a254aeb57060252ff2dbc7ab3bd26577daff1f874d1a4bee3dd19bae7a35b55bfa252b62d87af365bc","cypress-linux-x64-
3de529f50eafb2251381de866ff8199033b27c3ddba611a254aeb57060252ff2dbc7ab3bd26577daff1f874d1a4bee3dd19bae7a35b55bfa252b62d87af365bc"]
[Cypress CI Tests/start]   💬  ::debug::Checking zstd --version
[Cypress CI Tests/start]   💬  ::debug::Unable to locate executable file: zstd. Please verify either the file path exists or the file can be found within a directory specified by the PATH environment variable.
 Also check the file mode to verify the file is executable.
[Cypress CI Tests/start]   💬  ::debug::
[Cypress CI Tests/start]   💬  ::debug::getCacheEntry - Attempt 1 of 2 failed with error: Cache Service Url not found, unable to restore cache.
[Cypress CI Tests/start]   💬  ::debug::Unable to locate executable file: zstd. Please verify either the file path exists or the file can be found within a directory specified by the PATH environment variable.
 Also check the file mode to verify the file is executable.
[Cypress CI Tests/start]   💬  ::debug::
[Cypress CI Tests/start]   💬  ::debug::getCacheEntry - Attempt 1 of 2 failed with error: Cache Service Url not found, unable to restore cache.
[Cypress CI Tests/start]   💬  ::debug::getCacheEntry - Attempt 2 of 2 failed with error: Cache Service Url not found, unable to restore cache.
[Cypress CI Tests/start]   💬  ::debug::getCacheEntry - Attempt 2 of 2 failed with error: Cache Service Url not found, unable to restore cache.
| Restoring NPM cache error: getCacheEntry failed: Cache Service Url not found, unable to restore cache.
| Restoring Cypress cache error: getCacheEntry failed: Cache Service Url not found, unable to restore cache.
| [command]/usr/local/bin/npm ci
| npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
| npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random
for details.
| npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
| npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
| npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
| npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScript syntax/features added
| npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
| npm WARN deprecated rollup-plugin-babel@4.4.0: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-babel.
| npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
| npm WARN deprecated flatten@1.0.3: flatten is deprecated in favor of utility frameworks such as lodash.
| npm WARN deprecated babel-eslint@10.1.0: babel-eslint is now @babel/eslint-parser. This package will no longer receive updates.
| npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
| npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
| npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
| npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
| npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'
| npm WARN deprecated core-js@2.6.12: core-js@<3.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions c
```

cypress

See you in the next video

cypress

Extending Cypress

cypress

Extending Cypress

- Cypress can be extended with commands and tasks
- A custom command run in browser
  - Just like cy.get()
- A task runs in the Cypress backend Node process
  - Not limited by the browsers API

cypress

See you in the next video

cypress

Adding Custom Commands

cypress

Custom Commands

- Commands can be used to add new commands to the cy object
  - Or overwrite already existing ones
  - Normally added in cypress/support/commands.js
- Commands can either be parent, child or dual commands
  - Depending on how they are called

cypress

commands.js

JS commands.js M ×

cypress > support > JS commands.js > ...

```
27  Cypress.Commands.add('searchForAuthor', (author) ⇒ {
28    cy.get('select').select('inauthor');
29    cy.get('input').type(author);
30    cy.contains('button', 'Search').click();
31  });
```

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
88    [
89      {
90        name: 'Douglas Adams',
91        books: [
92          "The Hitchhiker's Guide to the Galaxy",
93          'So Long, and Thanks for All the Fish',
94        ],
95      },
96      {
97        name: 'James S. A. Corey',
98        books: ['The Expanse', 'Leviathan Falls'],
99      },
100   ].forEach((author) => {
101     author.books.forEach((book) => {
102       it(`Search for "${book}" by "${author.name}"`, () => {
103         cy.searchForAuthor(author.name);
104
105         cy.contains('.card-title', book).should('be.visible');
106       });
107     });
108   });
109 });
```

cypress

See you in the next video

cypress

Adding Child Commands

Adding Child Commands

- Child commands are chained of another command
  - Either a parent or another child command
- Add the prevSubject option
  - Either true for any previous child
  - Or element, document or window to specify the type of subject
- They receive the current subject as a parameter
  - And any additional parameters

cypress

commands.js

commands.js M

cypress > support > commands.js > ...

```
33  Cypress.Commands.add(
34    'checkSearchResultImage',
35    { prevSubject: 'element' },
36    (subject, title) ⇒ {
37      return cy
38        .wrap(subject)
39        .parents('.card')
40        .find('.card-img')
41        .should('be.visible')
42        .should('have.attr', 'alt', title);
43    }
44  );
```

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
100    ].forEach((author) => {
101      author.books.forEach((book) => {
102        it(`Search for "${book}" by "${author.name}"`, () => {
103          cy.searchForAuthor(author.name);
104
105          cy.contains('.card-title', book)
106            .should('be.visible')
107            .checkSearchResultImage(book);
108        });
109      });
110    });
111  });
```

cypress

See you in the next video

cypress

Writing Dual Commands

cypress

Dual Commands

- Dual commands can be used as both parent or child commands
  - Useful for some commands
- Specify that the prevSubject is optional
  - Will be undefined when used

commands.js

commands.js M

cypress > support > commands.js > ...

```
33 Cypress.Commands.add(
34   'checkSearchResultImage',
35   { prevSubject: ['element', 'optional'] },
36   (subject, title) ⇒ {
37     return (subject ? cy.wrap(subject) : cy.contains('.card-title', title))
38       .parents('.card')
39       .find('.card-img')
40       .should('be.visible')
41       .should('have.attr', 'alt', title);
42   }
43 );
```

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
100    ].forEach((author) => {
101      author.books.forEach((book) => {
102        it(`Search for "${book}" by "${author.name}"`, () => {
103          cy.searchForAuthor(author.name);
104
105          cy.contains('.card-title', book)
106            .should('be.visible')
107            .checkSearchResultImage(book);
108
109          cy.checkSearchResultImage(book).should('have.attr', 'src');
110        });
111      });
112    });
113  });
```

cypress

See you in the next video

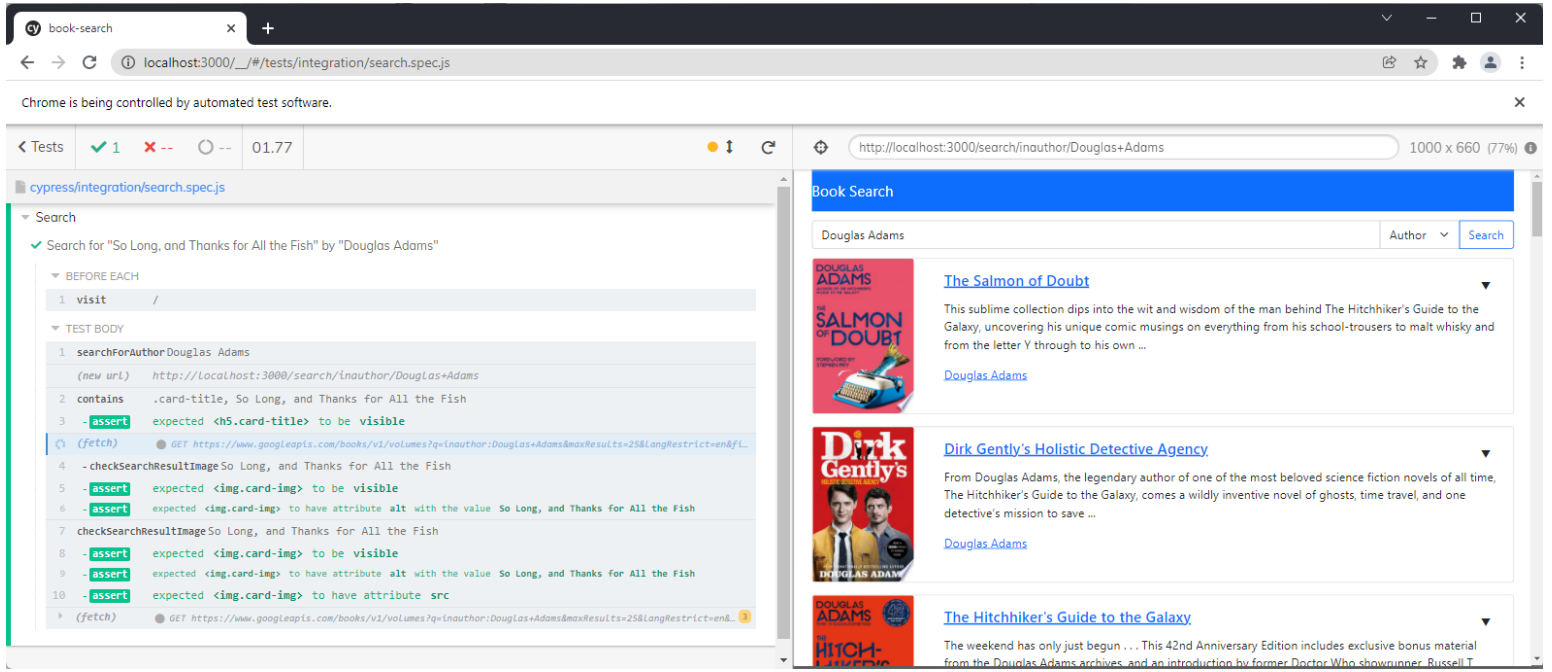
cypress

Logging Custom Commands

cypress

- Using Cypress.log() you can add custom log messages for commands
  - Recommended for custom commands
  - There is a cy.log() for simple messages
- Often you will want to suppress nested log messages
  - Use the { log: false } options

cypress


book-search
localhost:3000/__/#/tests/integration/search.spec.js
Chrome is being controlled by automated test software.
Tests
1
--
--
01.77
http://localhost:3000/search/inauthor/Douglas+Adams
1000 x 660 (77%)
cypress/integration/search.spec.js
Search
Search for "So Long, and Thanks for All the Fish" by "Douglas Adams"
BEFORE EACH
1 visit /
TEST BODY
1 searchForAuthor Douglas Adams
(new url) http://localhost:3000/search/inauthor/Douglas+Adams
2 contains .card-title, So Long, and Thanks for All the Fish
3 - assert expected <h5.card-title> to be visible
(fetch) GET https://www.googleapis.com/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&fi...
4 - checkSearchResultImage So Long, and Thanks for All the Fish
5 - assert expected <img.card-img> to be visible
6 - assert expected <img.card-img> to have attribute alt with the value So Long, and Thanks for All the Fish
7 checkSearchResultImage So Long, and Thanks for All the Fish
8 - assert expected <img.card-img> to be visible
9 - assert expected <img.card-img> to have attribute alt with the value So Long, and Thanks for All the Fish
10 - assert expected <img.card-img> to have attribute src
(fetch) GET https://www.googleapis.com/books/v1/volumes?q=inauthor:Douglas+Adams&maxResults=25&langRestrict=en&...
Book Search
Douglas Adams
Author
Search
The Salmon of Doubt
This sublime collection dips into the wit and wisdom of the man behind The Hitchhiker's Guide to the Galaxy, uncovering his unique comic musings on everything from his school-trousers to malt whisky and from the letter Y through to his own ...
Douglas Adams
Dirk Gently's Holistic Detective Agency
From Douglas Adams, the legendary author of one of the most beloved science fiction novels of all time, The Hitchhiker's Guide to the Galaxy, comes a wildly inventive novel of ghosts, time travel, and one detective's mission to save ...
Douglas Adams
The Hitchhiker's Guide to the Galaxy
The weekend has only just begun . . . This 42nd Anniversary Edition includes exclusive bonus material

commands.js

commands.js M

cypress > support > commands.js > ...

```
37  Cypress.Commands.add(
38    'checkSearchResultImage',
39    { prevSubject: ['element', 'optional'] },
40    (subject, title) => {
41      Cypress.log({
42        name: 'checkSearchResultImage',
43        message: title,
44        type: subject ? 'child' : 'parent',
45      });
46
47      return (
48        subject
49          ? cy.wrap(subject, { log: false })
50          : cy.contains('.card-title', title, { log: false })
51      )
52        .parents('.card', { log: false })
53        .find('.card-img', { log: false })
54        .should('be.visible')
55        .should('have.attr', 'alt', title);
56    }
57  );
```

cypress

See you in the next video

cypress

Better Command Logging

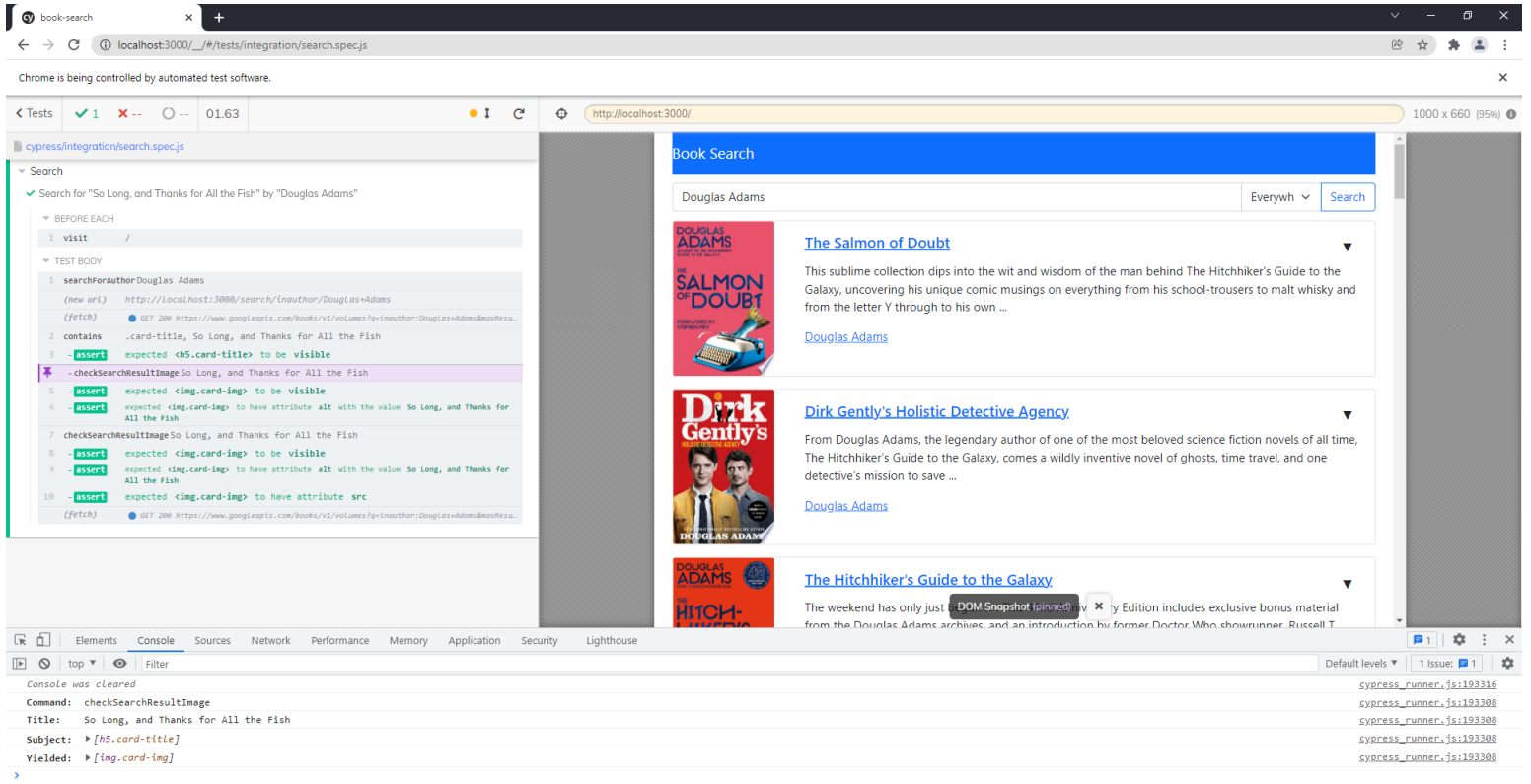
cypress

Better Command Logging

- Cypress can show additional debug information
  - When you open the dev tools console and click on a command
- The Cypress.log() function will accept a consoleProps() function
  - The result is printed to the dev tools console on click

cypress

Console Output



commands.js

commands.js M

cypress > support > commands.js > ...

```
37  Cypress.Commands.add(
38    'checkSearchResultImage',
39    { prevSubject: ['element', 'optional'] },
40    (subject, title) => {
41      const props = {
42        title,
43        subject: subject?.toArray(),
44      };
45
46      Cypress.log({
47        type: subject ? 'child' : 'parent',
48        consoleProps: () => props,
49      });
50
51      return (
52        subject
53          ? cy.wrap(subject, { log: false })
54          : cy.contains('.card-title', title, { log: false })
55      )
56        .parents('.card', { log: false })
57        .find('.card-img', { log: false })
58        .then((el) => {
59          props.yielded = el.toArray();
60          return el;
61        })
62        .should('be.visible')
63        .should('have.attr', 'alt', title);
64    }
```

cypress

See you in the next video

cypress

Adding Custom Tasks

cypress

Custom Tasks

- Task are functions that run in the node environment of Cypress
  - Unlike a command that executes in the browser
- Tasks always need to return a value or a promise of a value
  - Or null in the case of no explicit value
  - Arguments and result values need to be serializable
- Tasks can be started with the cy.task() command
  - The result is passed to a then() handler
- Each task has a default timeout of taskTimeout (1 minute)
  - But that can be overridden using the timeout option

cypress

plugins/index.js

search.spec.js M | index.js M

cypress > plugins > index.js > ...

```
22  module.exports = (on, config) => {
23    // `on` is used to hook into various events Cypress emits
24    // `config` is the resolved Cypress config
25    on('task', {
26      log(message) {
27        console.log(`\u001B[31m${message}\u001B[39m`);
28        return null;
29      },
30      async readFileOrDefault({ filePath, defaultValue = null }) {
31        const fullPath = resolve(filePath);
32
33        try {
34          await access(fullPath);
35
36          const data = await readFile(fullPath, 'utf8');
37          return JSON.parse(data);
38        } catch (err) {}
39
40        return defaultValue;
41      },
42    });
43  };
```

cypress

search.spec.js

search.spec.js M × | index.js M

cypress > integration > search.spec.js > ...

```
114    it.only('Read an optional file', () => {
115      cy.readFile('cypress/fixtures/example.json').then(cy.log);
116      // cy.readFile('not-there.json').then(cy.log);
117
118      cy.task('readFileOrDefault', {
119        filePath: 'cypress/fixtures/example.json',
120      }).then(cy.log);
121
122      cy.task('readFileOrDefault', {
123        filePath: 'not-there.json',
124        defaultValue: { notFound: true },
125      }).then(cy.log);
126    });
127  });
```

cypress

See you in the next video

cypress

Combining Tasks & Commands

cypress

Combining Tasks & Commands

- Tasks are great to execute Node.js code
  - But the cy.task() syntax with parameters can be akward
- Use custom commands to make tasks easier to use
  - Including IntelliSense in Visual Studio Code
  - And type checking when using TypeScript

cypress

commands.js

commands.js M

cypress > support > commands.js > ...

```
67 Cypress.Commands.add('readFileOrDefault', (filePath, defaultValue) ⇒ {
68   Cypress.log();
69
70   return cy.task(
71     'readFileOrDefault',
72     {
73       filePath,
74       defaultValue,
75     },
76     { log: false }
77   );
78 });
```

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
114   it.only('use a task', () => {
115     cy.log('⟹ Log this message in Cypress. ⟸');
116     cy.task('log', '⟹ Log this message in the Node console. ⟸');
117
118     cy.readFile('cypress/fixtures/example.json').then(cy.log);
119     // cy.readFile('not-there.json').then(cy.log);
120
121     cy.readFileOrDefault('cypress/fixtures/example.json').then(cy.log);
122     cy.readFileOrDefault('not-there.json', { notFound: true }).then(cy.log);
123   });
124 });
```

cypress

See you in the next video

cypress

Cypress Testing Library

cypress

Cypress Testing Library

- Test your UI components in **a more user-centric way**
  - Users don't use CSS queries
- Allows **better queries** within Cypress
  - Queries are often bases on accessibility
- **DOM/React/Vue/Angular/Svelte Testing Library**
  - Also usable **with Jest**
  - Using the same query functions

cypress

See you in the next video

cypress

Tip & Tricks

cypress

Tip & Tricks

cypress

See you in the next video

cypress

IntelliSense in Visual Studio Code

cypress

IntelliSense

- Add a reference to a package types using **triple slash directives**
  - /// <reference types="cypress" />

cypress

search.spec.js 4

cypress > integration > search.spec.js > ...

```
1  /// <reference types="cypress" />
2
3  describe('Search', () => {
4    beforeEach(() => {
5      cy.visit('/');
6    });
```

cypress

See you in the next video

cypress

Configuring ESLint

cypress

Configuring ESLint

- Install the eslint-plugin-cypress NPM package
- Extend the **plugin:cypress/recommended** in your ESLint config

cypress

package.json

package.json M

package.json > ...

```
23    "eslintConfig": {
24      "extends": [
25        "react-app",
26        "plugin:cypress/recommended"
27      ]
28    },
29    "browserslist": {
30      "production": [
31        ">0.2%",
32        "not dead",
33        "not op_mini all"
34      ],
35      "development": [
36        "last 1 chrome version",
37        "last 1 firefox version",
38        "last 1 safari version"
39      ]
40    },
41    "devDependencies": {
42      "@testing-library/jest-dom": "^5.14.1",
43      "@testing-library/react": "^12.1.0",
44      "@testing-library/user-event": "^13.2.1",
45      "@types/jest": "^27.0.1",
46      "@types/node": "^16.9.1",
47      "@types/react": "^17.0.20",
48      "@types/react-dom": "^17.0.9",
49      "@types/react-router-dom": "^5.1.8",
50      "cypress": "^8.4.0",
51      "eslint-plugin-cypress": "^2.12.1",
```

cypress

See you in the next video

cypress

Faking Math.random()

cypress

Faking Math.random()

- The cy.stub() command will let fake an existing function
  - Including native browser functions
- It returns a Sinon.js stub
- A cy.visit() reloads the page and resets the stub
- https://docs.cypress.io/api/commands/stub

cypress

random.spec.js

```
JS random.spec.js U ×
cypress > integration > JS random.spec.js > ...
1  /// <reference types='cypress' />
2
3  describe('Random numbers', () => {
4    beforeEach(() => {
5      cy.visit('/random.html');
6
7      const randomQueue = [
8        0.1750883340521796, 0.9007112009326497, 0.0019656171919002485,
9        0.740324897488438, 0.555555555555555, 0.666666666666666,
10     ];
11
12     cy.window().then((win) => {
13       cy.stub(win.Math, 'random').callsFake(() => {
14         const rnd = randomQueue.shift();
15         randomQueue.push(rnd);
16         return rnd;
17       });
18     });
19   });
20
21   it('Make sure they are there', () => {
22     cy.get('#btn-generate').click();
23
24     cy.contains('Random number 1: 0.1750883340521796').should('be.visible');
25     cy.contains('Random number 2: 0.9007112009326497').should('be.visible');
26   });
27 });
```

cypress

See you in the next video

cypress

Overriding the configuration

cypress

Overriding the configuration

- You can override the Cypress runtime configuration as needed
  - Using the --config command line option
- Useful to run the Cypress tests against another base URL
  - Like an acceptance environment

cypress

package.json

package.json M

package.json > ...

```
15    "scripts": {
16      "start": "react-scripts start",
17      "build": "react-scripts build",
18      "test": "react-scripts test",
19      "prettier": "prettier --write {src,public}/**/*.{js,ts,jsx,tsx,css,json,html}",
20      "cypress:open": "cypress open",
21      "cypress:run": "cypress run --record",
22      "cypress:run:azure": "cypress run --config baseUrl=https://book-search.azureedge.net/"
23    },
```

cypress

See you in the next video

cypress

Data driven tests

cypress

Data driven tests

- Writing tests that validate the same sort of thing is common
  - The only differences are the values used
- Mocha has no `describe.each()` or `it.each()` by default
  - Easy to add using cypress-each
- Can also easy to do with a simple JavaScript loop

cypress

search.spec.js

search.spec.js M

cypress > integration > search.spec.js > ...

```
88    [
89      {
90        name: 'Douglas Adams',
91        books: [
92          "The Hitchhiker's Guide to the Galaxy",
93          'So Long, and Thanks for All the Fish',
94        ],
95      },
96      {
97        name: 'James S. A. Corey',
98        books: ['The Expanse', 'Leviathan Falls'],
99      },
100   ].forEach((author) => {
101     author.books.forEach((book) => {
102       it(`Search for "${book}" by "${author.name}"`, () => {
103         cy.get('select').select('inauthor');
104         cy.get('input').type(author.name);
105         cy.contains('button', 'Search').click();
106
107         cy.contains('.card-title', book).should('be.visible');
108       });
109     });
110   });
111 });
```

cypress

Data driven tests

book-search

localhost:3000/__/#/tests/integration/search.spec.js

Chrome is being controlled by automated test software.

Tests 10 -- -- 23.61

cypress/integration/search.spec.js

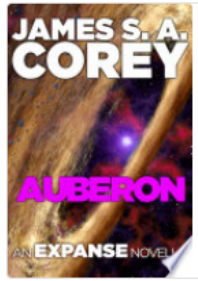
Search

- For books by Douglas Adams
- intercept with a StringMatcher
- intercept with a RouteMatcher
- wait for fetch()
- intercept and fake the response
- intercept and use the response
- Search for "The Hitchhiker's Guide to the Galaxy" by "Douglas Adams"
- Search for "So Long, and Thanks for All the Fish" by "Douglas Adams"
- Search for "The Expanse" by "James S. A. Corey"
- Search for "Leviathan Falls" by "James S. A. Corey"

http://localhost:3000/search/inauthor/James+S%E2%80%A2+A%E2%80%

Book Search

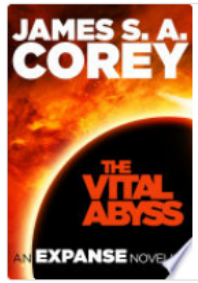
James S. A. Corey



Auberon

A novella set in the u
explores a new and a

James S. A. Corey



The Vital Abyss

. . Perhaps one of the
it's supposed to be' \

James S. A. Corey

cypress

See you in the next video

cypress

Maurice de Beijer

@mauricedb

maurice.de.beijer
@gmail.com

cypress