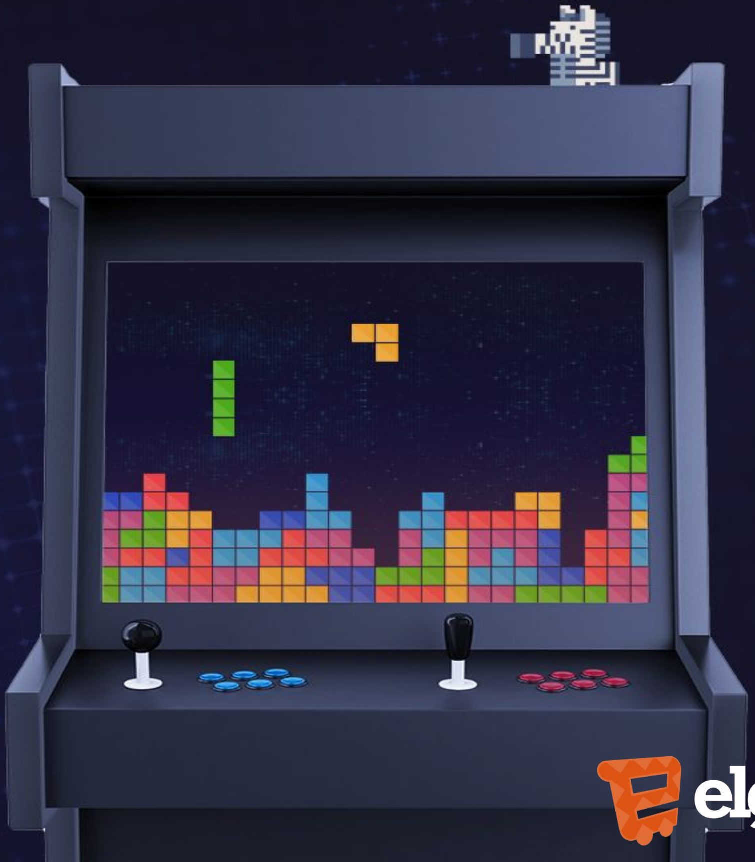
Cypress for Magento 2

Reacticon 4

October 11th, 2021



elgentos
ecommerce solutions

Peter Jaap Blaakmeer

@PeterJaap

CTO @ elgentos (Groningen, NL)

What is Cypress?

- Testing framework
- Javascript-based
- Lives right in your browser
  - Knows about the DOM
  - Can hook into just about everything
  - Retry-ability
  - Automatic waiting on XHR’s, animations, etc - no more weird timing issues
- This all means; it’s fast!
- MFTF is a pain to work with

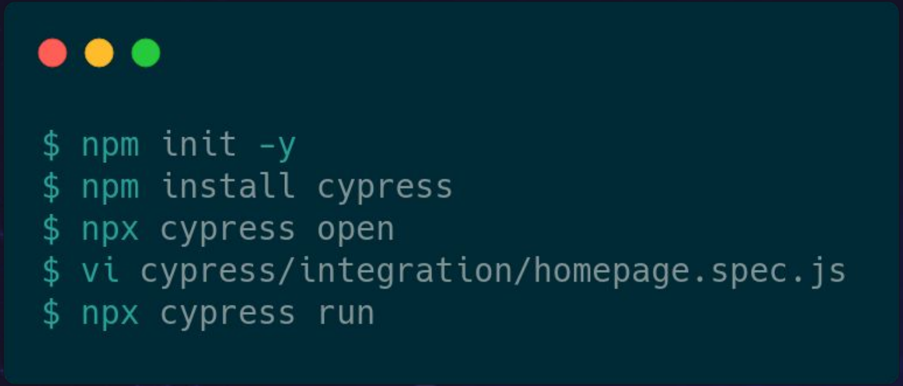
What kind of testing can Cypress do?

- End to end testing; frontend & backend
- Integration testing; test how your code functions with third party code
- Unit testing; test a very specific part of your code in isolation
- Load testing; testing how your store handles a large influx of users

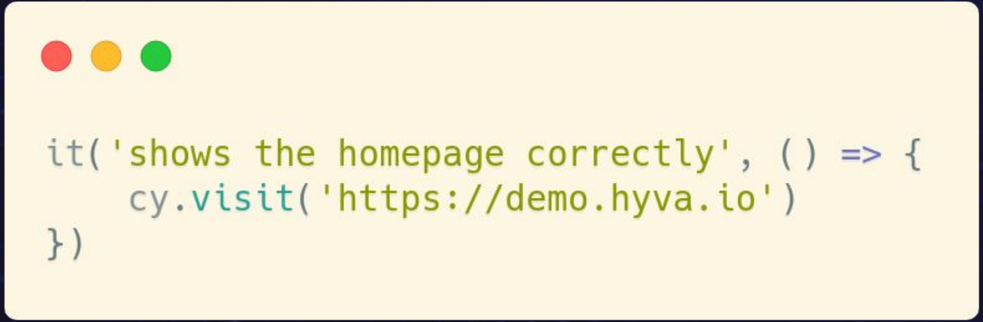
We'll focus on the first two and let's write some code.

elgentos
ecommerce solutions

Cypress installation & first test



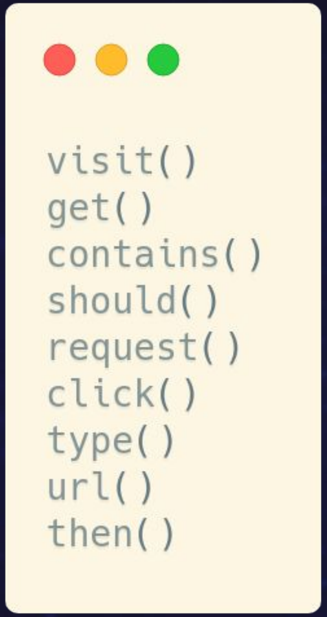
```
$ npm init -y
$ npm install cypress
$ npx cypress open
$ vi cypress/integration/homepage.spec.js
$ npx cypress run
```



```
it('shows the homepage correctly', () => {
    cy.visit('https://demo.hyva.io')
})
```

Commands

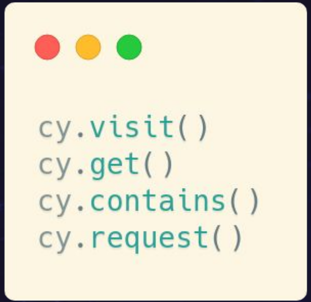
Most often used;



```
visit()
get()
contains()
should()
request()
click()
type()
url()
then()
```

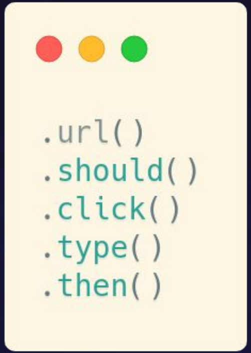
Parent / child commands

Parent commands



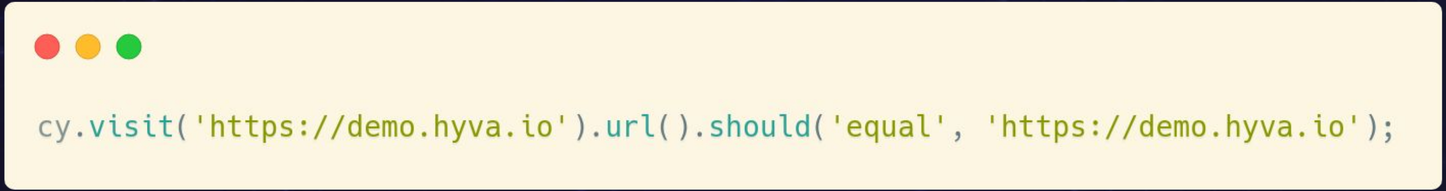
```
cy.visit()
cy.get()
cy.contains()
cy.request()
```

Child commands

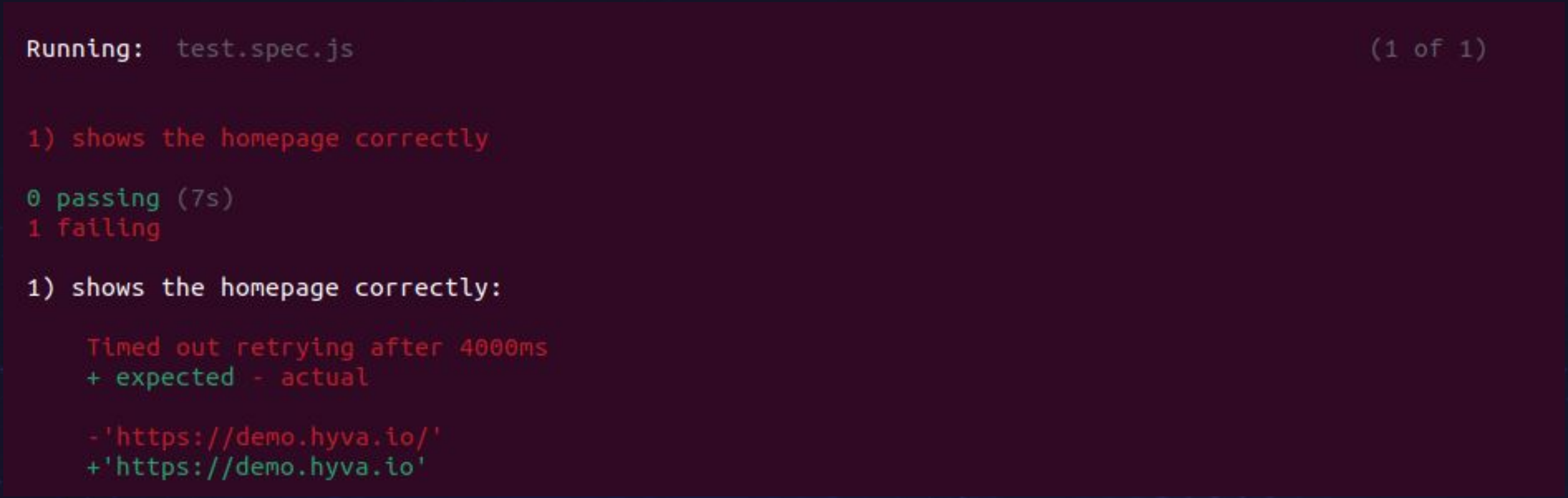


```
.url()
.should()
.click()
.type()
.then()
```

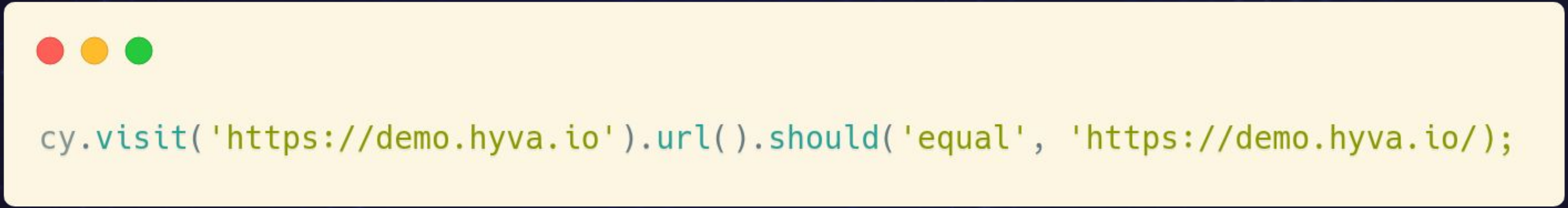
Child commands are chained off parent commands, like this;



```
cy.visit('https://demo.hyva.io').url().should('equal', 'https://demo.hyva.io');
```

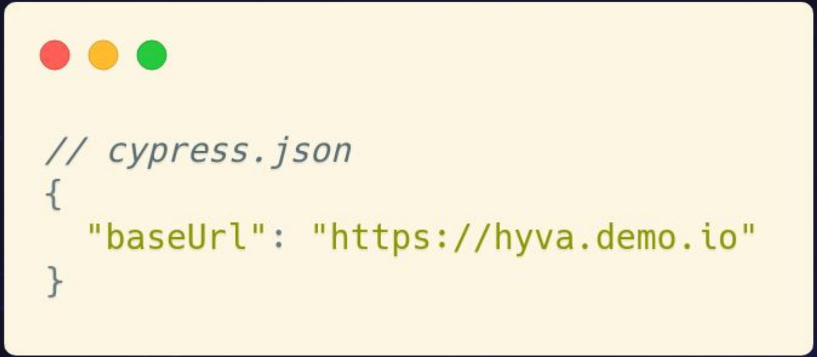


```
Running:  test.spec.js                                                   (1 of 1)


1) shows the homepage correctly

0 passing (7s)
1 failing

1) shows the homepage correctly:

    Timed out retrying after 4000ms
    + expected - actual

    -'https://demo.hyva.io/'
    +'https://demo.hyva.io'
```



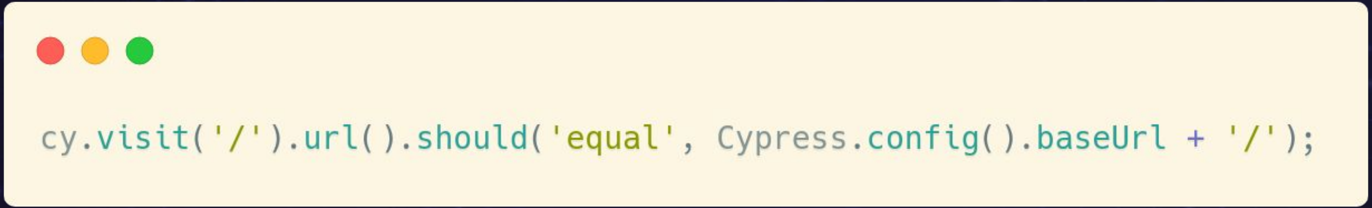
```
cy.visit('https://demo.hyva.io').url().should('equal', 'https://demo.hyva.io/);
```

Configure baseUrl

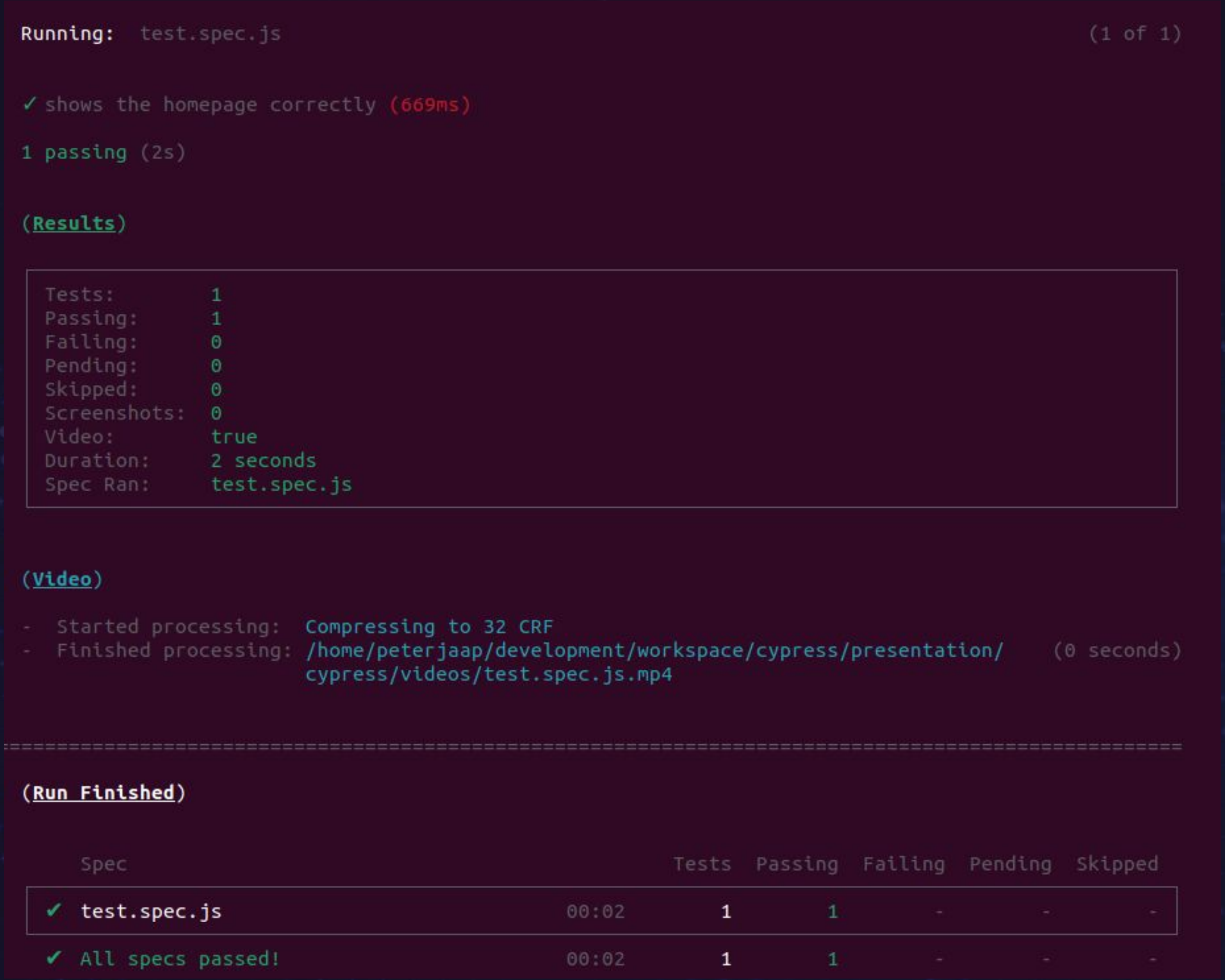


```
// cypress.json
{
  "baseUrl": "https://hyva.demo.io"
}
```

Adjust test, which will now always succeed;



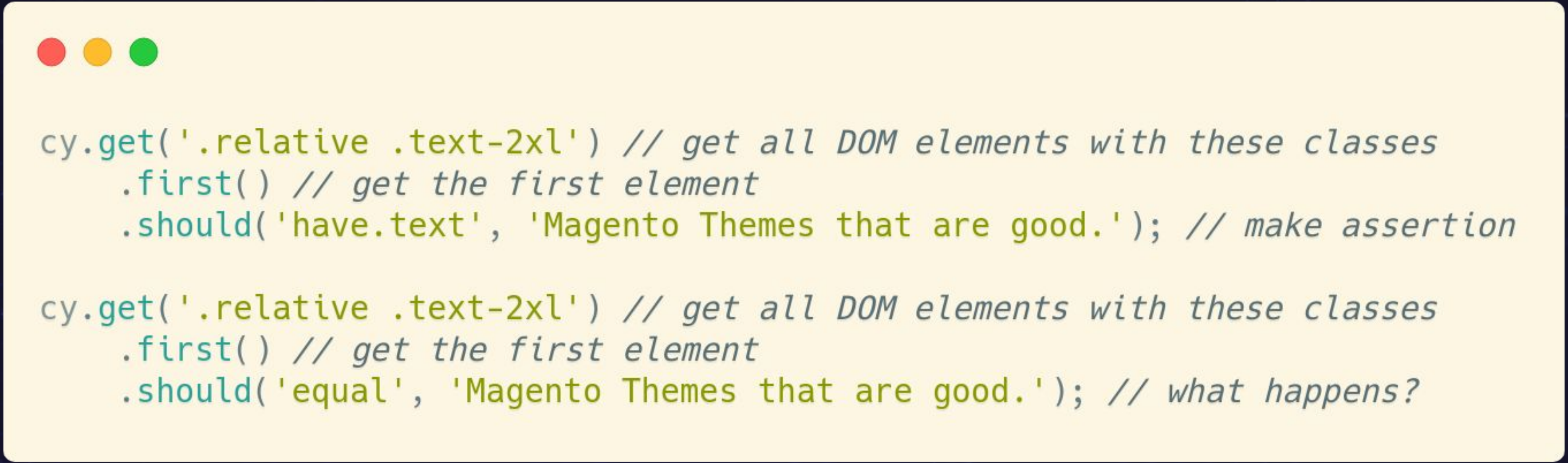
```
cy.visit('/').url().should('equal', Cypress.config().baseUrl + '/');
```



```
Running:  test.spec.js                                                                   (1 of 1)


  ✓ shows the homepage correctly (669ms)

  1 passing (2s)


  (Results)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Tests:        1                                                                                │
  │ Passing:      1                                                                                │
  │ Failing:      0                                                                                │
  │ Pending:      0                                                                                │
  │ Skipped:      0                                                                                │
  │ Screenshots:  0                                                                                │
  │ Video:        true                                                                             │
  │ Duration:     2 seconds                                                                        │
  │ Spec Ran:     test.spec.js                                                                     │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘


  (Video)

  -  Started processing:  Compressing to 32 CRF
  -  Finished processing: /home/peterjaap/development/workspace/cypress/presentation/    (0 seconds)
                          cypress/videos/test.spec.js.mp4


====================================================================================================

  (Run Finished)


       Spec                                              Tests  Passing  Failing  Pending  Skipped
  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ ✔  test.spec.js                            00:02        1        1        -        -        - │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘
    ✔  All specs passed!                       00:02        1        1        -        -        -
```

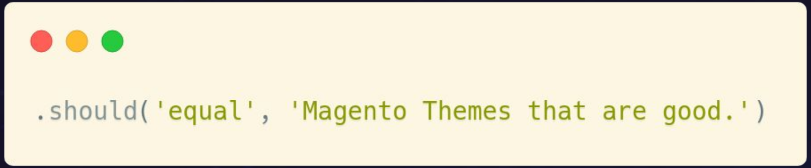
elgentos
ecommerce solutions

Let’s cy.get() an element



```
cy.get('.relative .text-2xl') // get all DOM elements with these classes
    .first() // get the first element
    .should('have.text', 'Magento Themes that are good.'); // make assertion

cy.get('.relative .text-2xl') // get all DOM elements with these classes
    .first() // get the first element
    .should('equal', 'Magento Themes that are good.'); // what happens?
```

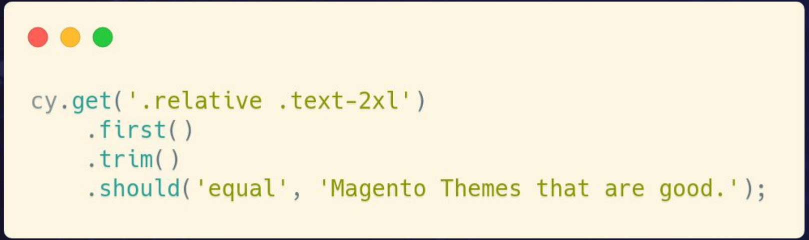
Asserting equality



```
.should('equal', 'Magento Themes that are good.')
```

This tests whether the content of the element **exactly** matches the string. But HTML is messy!

Another option is to trim the whitespace and then assert equality;

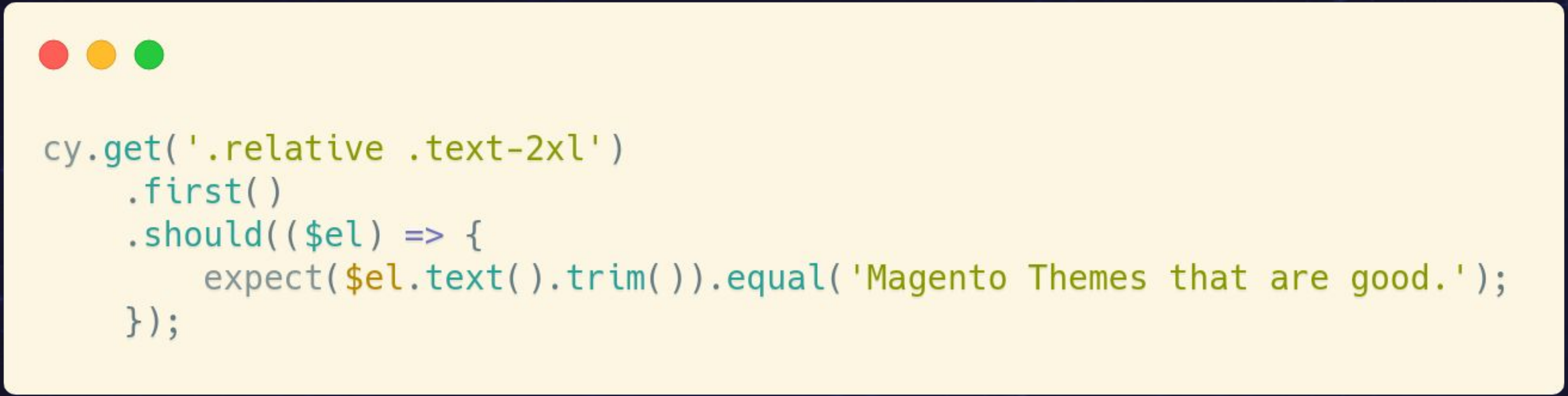


```
cy.get('.relative .text-2xl')
    .first()
    .trim()
    .should('equal', 'Magento Themes that are good.');
```

Looks good, right?

Cypress yields instead of returns

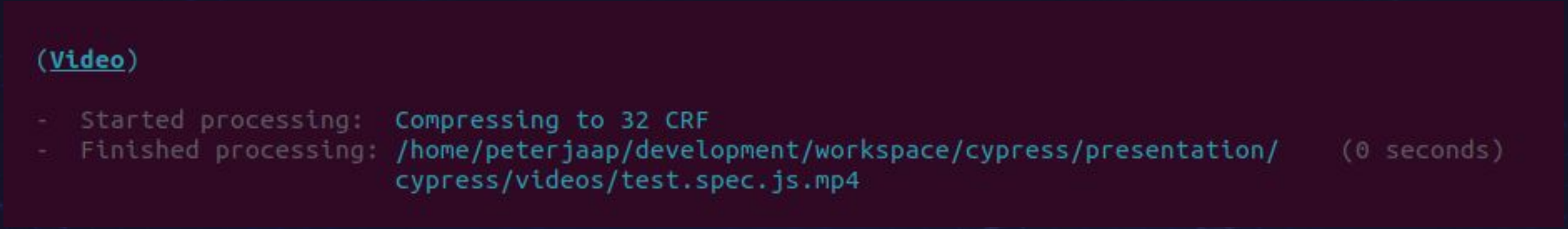
Looks good, but it isn’t. Cypress is asynchronous, so it yields. We’ll need a callback;



```
cy.get('.relative .text-2xl')
    .first()
    .should(($el) => {
        expect($el.text().trim()).equal('Magento Themes that are good.');
    });
```

Looks familiar? Cypress comes with jQuery by default. You can even access it directly through Cypress.$

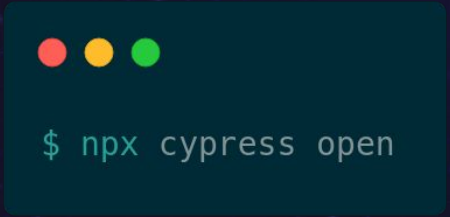
Wait what, video?!



```
(Video)

-  Started processing:  Compressing to 32 CRF
-  Finished processing: /home/peterjaap/development/workspace/cypress/presentation/    (0 seconds)
                        cypress/videos/test.spec.js.mp4
```

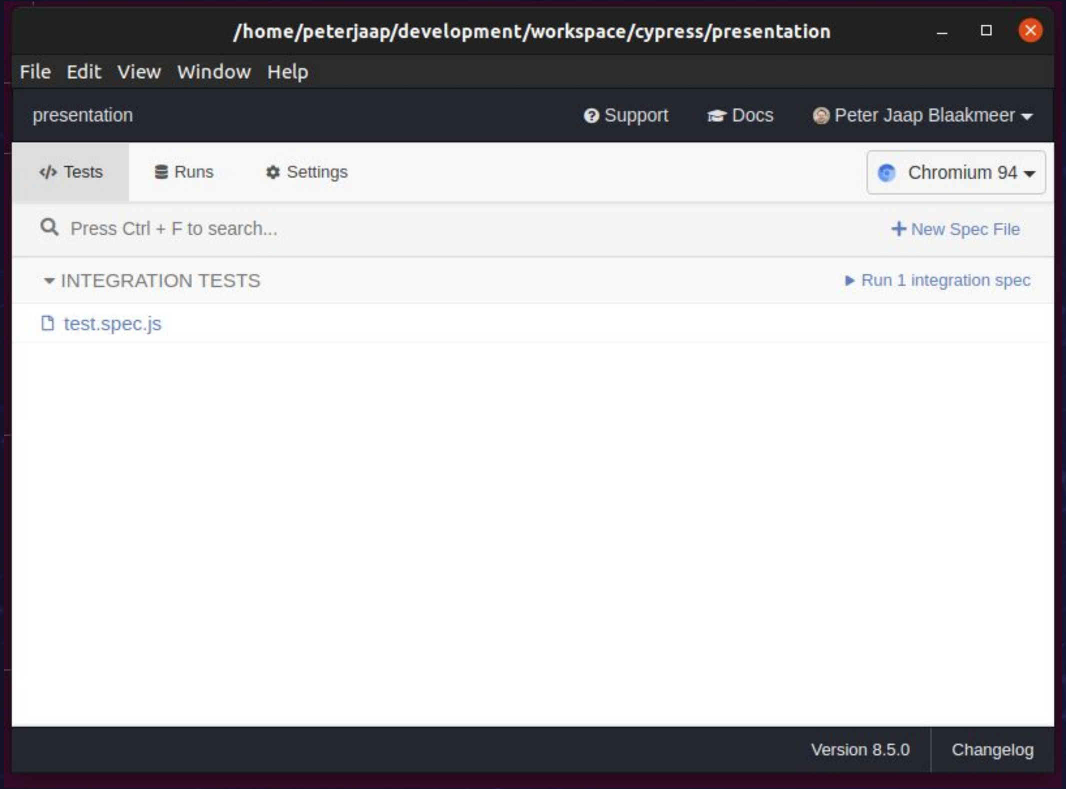
Cypress automatically records videos and stores them in cypress/videos.

But, there's realtime too!



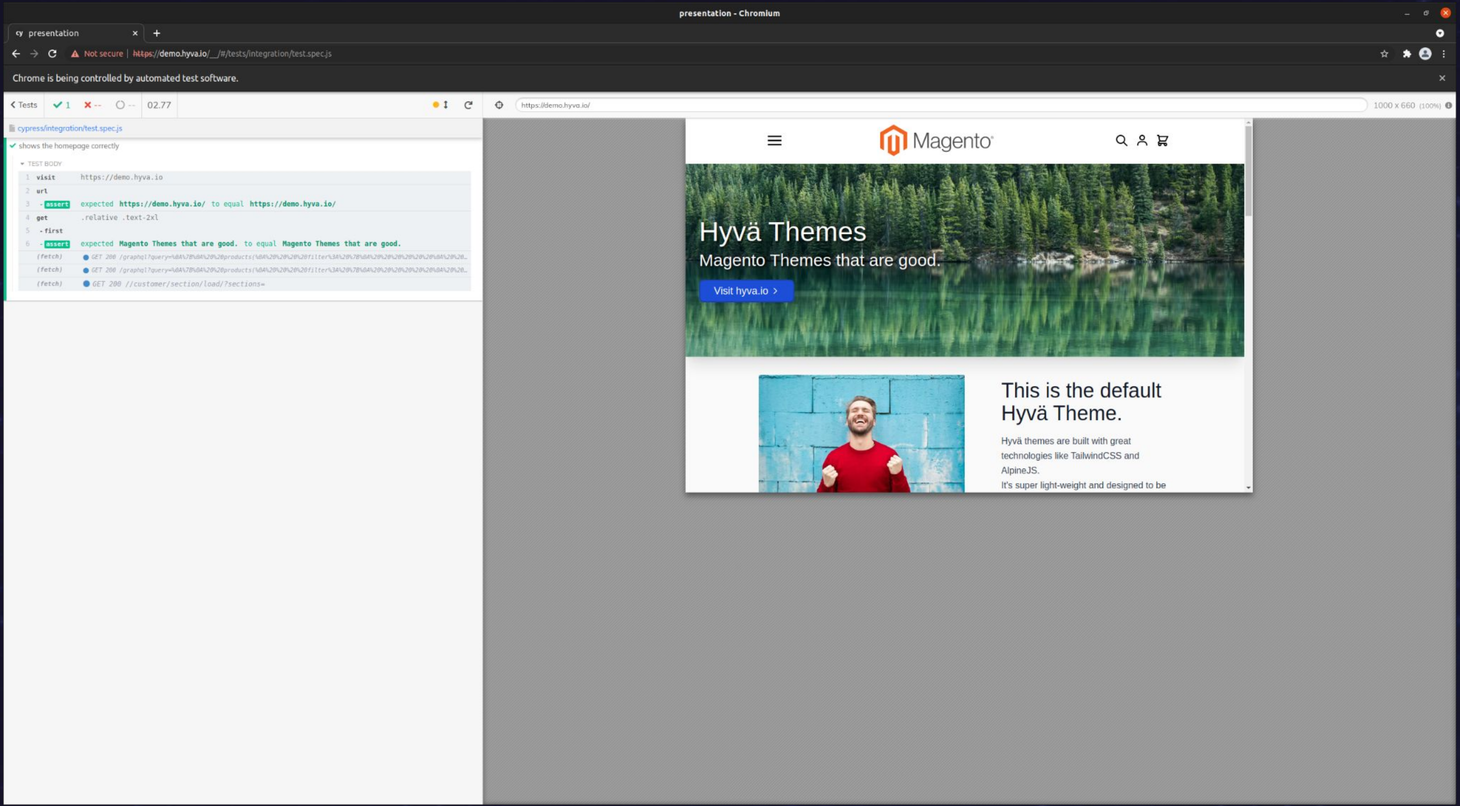
```
$ npx cypress open
```

Cypress Test Runner



/home/peterjaap/development/workspace/cypress/presentation

Cypress Test Runner



presentation - Chromium
presentation
Not secure | https://demo.hyva.io/__/#/tests/integration/test.spec.js

Chrome is being controlled by automated test software.

Tests ✓ 1 ✕ -- ○ -- 02.77
https://demo.hyva.io/
1000 x 660 (100%)

cypress/integration/test.spec.js

✓ shows the homepage correctly

TEST BODY

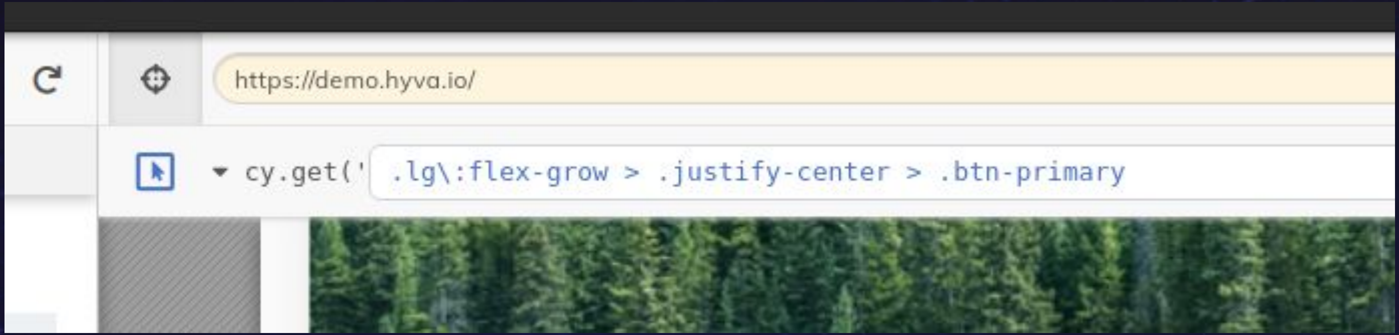
```
1 visit     https://demo.hyva.io
2 url
3  -assert  expected https://demo.hyva.io/ to equal https://demo.hyva.io/
4 get       .relative .text-2xl
5  -first
6  -assert  expected Magento Themes that are good. to equal Magento Themes that are good.
  (fetch)   ● GET 200 /graphql?query=%0A%7B%0A%20%20products(%0A%20%20%20%20filter%3A%20%7B%0A%20%20%20%20%20%20%0A%20%20…
  (fetch)   ● GET 200 /graphql?query=%0A%7B%0A%20%20products(%0A%20%20%20%20filter%3A%20%7B%0A%20%20%20%20%20%20%0A%20%20…
  (fetch)   ● GET 200 //customer/section/load/?sections=
```

This is the default Hyvä Theme.

Hyvä themes are built with great technologies like TailwindCSS and AlpineJS.

It's super light-weight and designed to be

Selector Playground

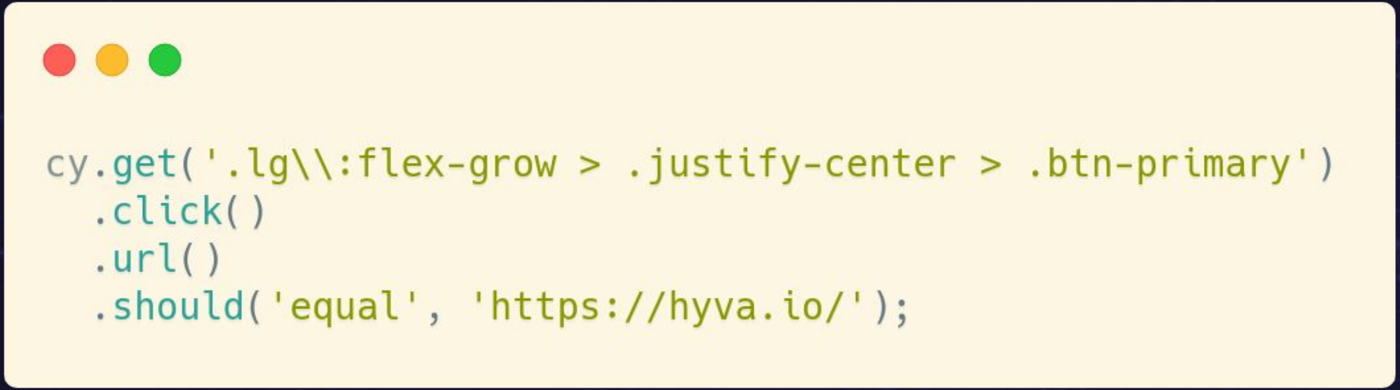


https://demo.hyva.io/

```
cy.get(' .lg\:flex-grow > .justify-center > .btn-primary
```

Time Travel

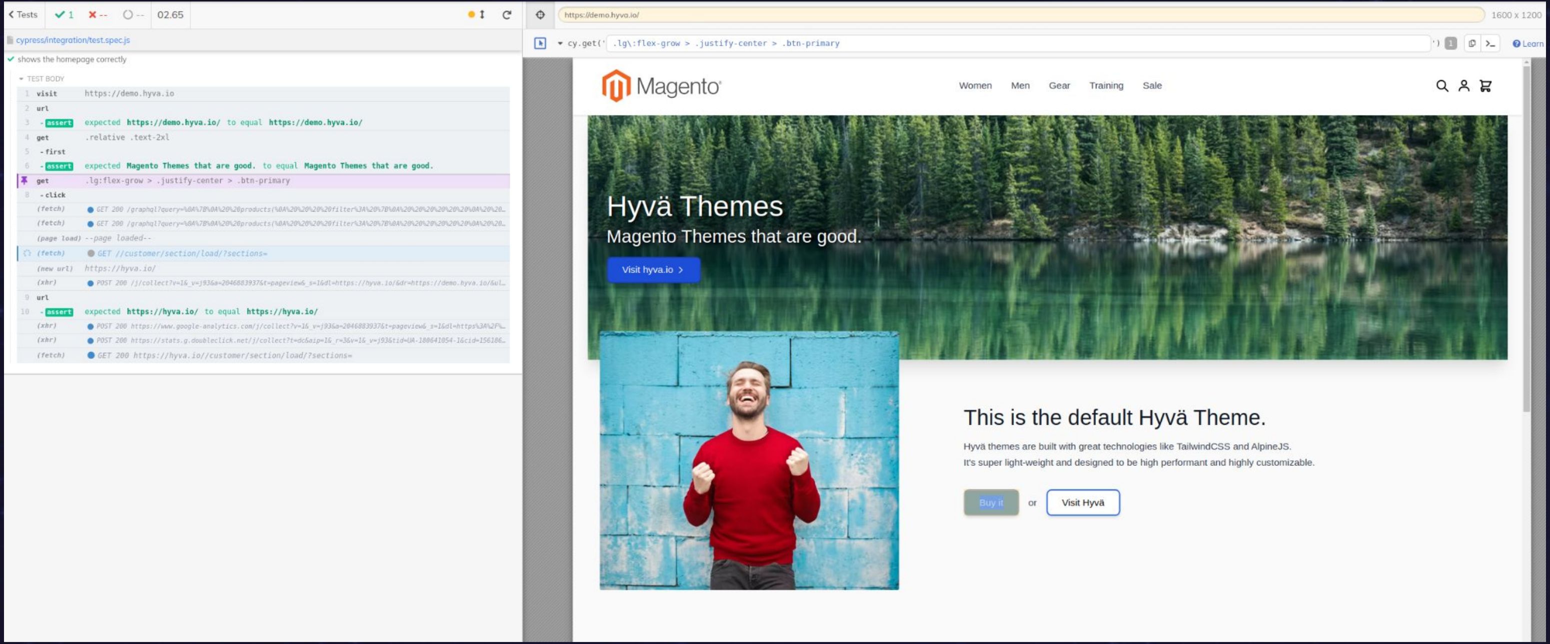
Let’s navigate somewhere;



```
cy.get('.lg\\:flex-grow > .justify-center > .btn-primary')
  .click()
  .url()
  .should('equal', 'https://hyva.io/');
```

Time Travel

And travel through time by hovering;



cypress/integration/test.spec.js

✓ shows the homepage correctly

TEST BODY

```
1  visit        https://demo.hyva.io
2  url
3   - assert    expected https://demo.hyva.io/ to equal https://demo.hyva.io/
4  get          .relative .text-2xl
5   - first
6   - assert    expected Magento Themes that are good. to equal Magento Themes that are good.
   get          .lg:flex-grow > .justify-center > .btn-primary
8   - click
   (fetch)      ● GET 200 /graphql?query=%0A%7B%0A%20%20products(%0A%20%20%20%20filter%3A%20%7B%0A%20%20%20%20%20%20%0A%20%20…
   (fetch)      ● GET 200 /graphql?query=%0A%7B%0A%20%20products(%0A%20%20%20%20filter%3A%20%7B%0A%20%20%20%20%20%20%0A%20%20…
   (page load)  --page loaded--
   (fetch)      ● GET //customer/section/load/?sections=
   (new url)    https://hyva.io/
   (xhr)        ● POST 200 /j/collect?v=1&_v=j93&a=2046883937&t=pageview&_s=1&dl=https://hyva.io/&dr=https://demo.hyva.io/&ul…
9  url
10  - assert    expected https://hyva.io/ to equal https://hyva.io/
   (xhr)        ● POST 200 https://www.google-analytics.com/j/collect?v=1&_v=j93&a=2046883937&t=pageview&_s=1&dl=https%3A%2F%…
   (xhr)        ● POST 200 https://stats.g.doubleclick.net/j/collect?t=dc&aip=1&_r=3&v=1&_v=j93&tid=UA-180641054-1&cid=156186…
   (fetch)      ● GET 200 https://hyva.io//customer/section/load/?sections=
```

```
cy.get('.lg\:flex-grow > .justify-center > .btn-primary') 1
```

Hyvä Themes

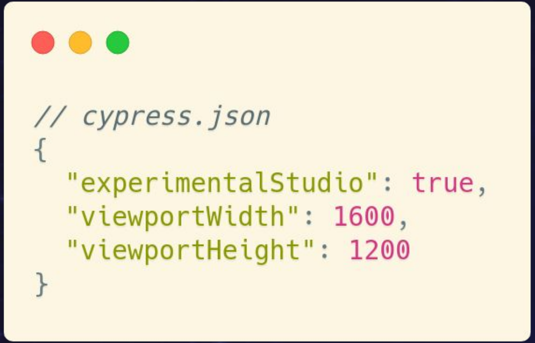
Magento Themes that are good.

This is the default Hyvä Theme.

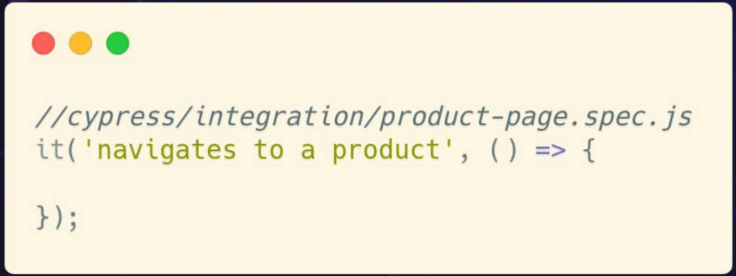
Hyvä themes are built with great technologies like TailwindCSS and AlpineJS.
It's super light-weight and designed to be high performant and highly customizable.

Cypress Studio

Some configuration:

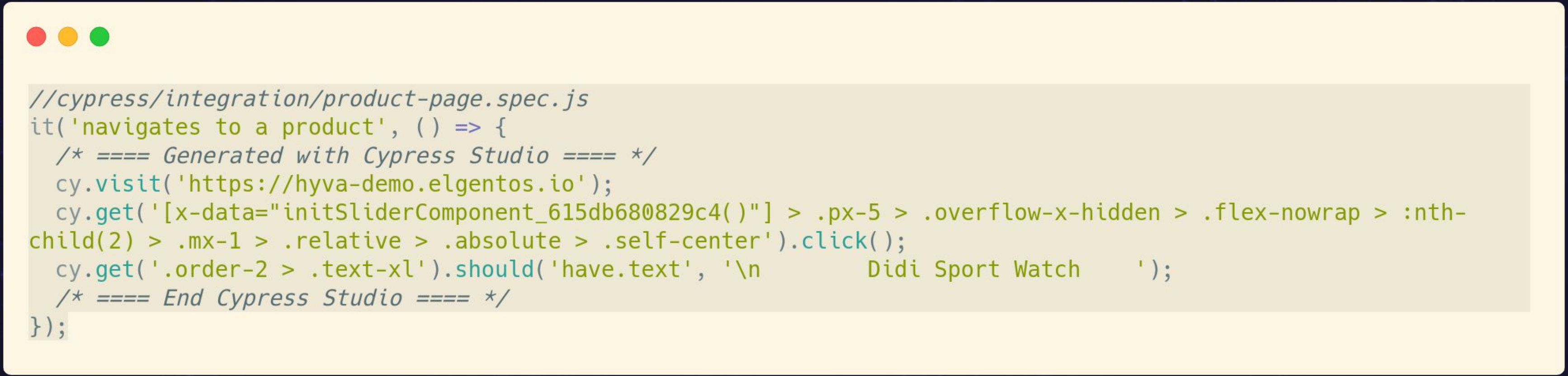


```
// cypress.json
{
  "experimentalStudio": true,
  "viewportWidth": 1600,
  "viewportHeight": 1200
}
```



```
//cypress/integration/product-page.spec.js
it('navigates to a product', () => {

});
```

Cypress Studio



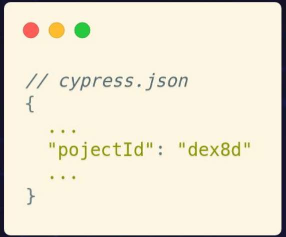
```
//cypress/integration/product-page.spec.js
it('navigates to a product', () => {
  /* ==== Generated with Cypress Studio ==== */
  cy.visit('https://hyva-demo.elgentos.io');
  cy.get('[x-data="initSliderComponent_615db680829c4()"] > .px-5 > .overflow-x-hidden > .flex-nowrap > :nth-child(2) > .mx-1 > .relative > .absolute > .self-center').click();
  cy.get('.order-2 > .text-xl').should('have.text', '\n        Didi Sport Watch    ');
  /* ==== End Cypress Studio ==== */
});
```

Cypress Studio

- Terrible test code
- Useful to quickly generate a test
- Always abstract the extremely specific selectors

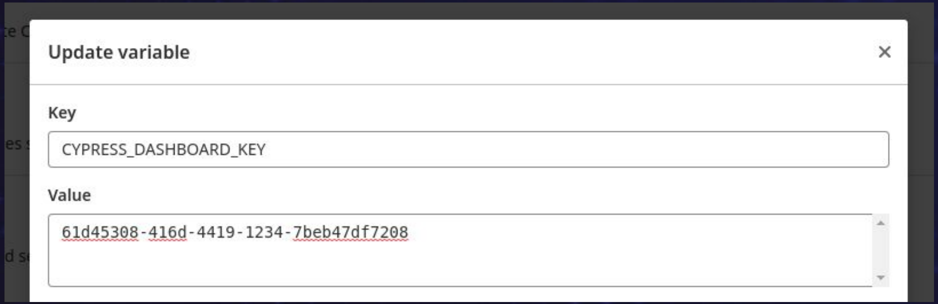
Cypress Dashboard

Configure Cypress through cypress.json



```
// cypress.json
{
  ...
  "pojectId": "dex8d"
  ...
}
```

Set an environment variable with your dashboard key in your CI



Update variable

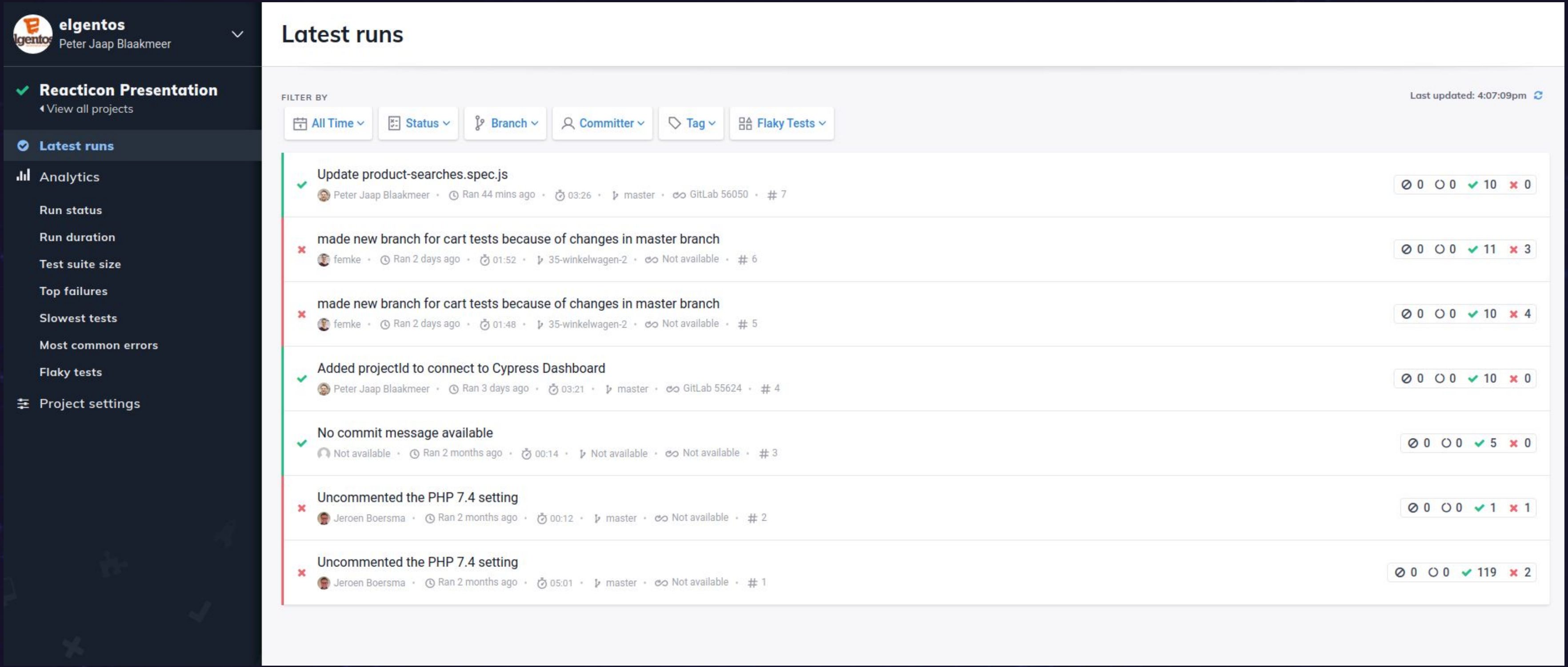
Key

CYPRESS_DASHBOARD_KEY

Value

61d45308-416d-4419-1234-7beb47df7208

Cypress Dashboard



Latest runs

Last updated: 4:07:09pm

✓ Update product-searches.spec.js
Peter Jaap Blaakmeer · Ran 44 mins ago · 03:26 · master · GitLab 56050 · # 7
⊘ 0 ○ 0 ✓ 10 ✗ 0

✗ made new branch for cart tests because of changes in master branch
femke · Ran 2 days ago · 01:52 · 35-winkelwagen-2 · Not available · # 6
⊘ 0 ○ 0 ✓ 11 ✗ 3

✗ made new branch for cart tests because of changes in master branch
femke · Ran 2 days ago · 01:48 · 35-winkelwagen-2 · Not available · # 5
⊘ 0 ○ 0 ✓ 10 ✗ 4

✓ Added projectId to connect to Cypress Dashboard
Peter Jaap Blaakmeer · Ran 3 days ago · 03:21 · master · GitLab 55624 · # 4
⊘ 0 ○ 0 ✓ 10 ✗ 0

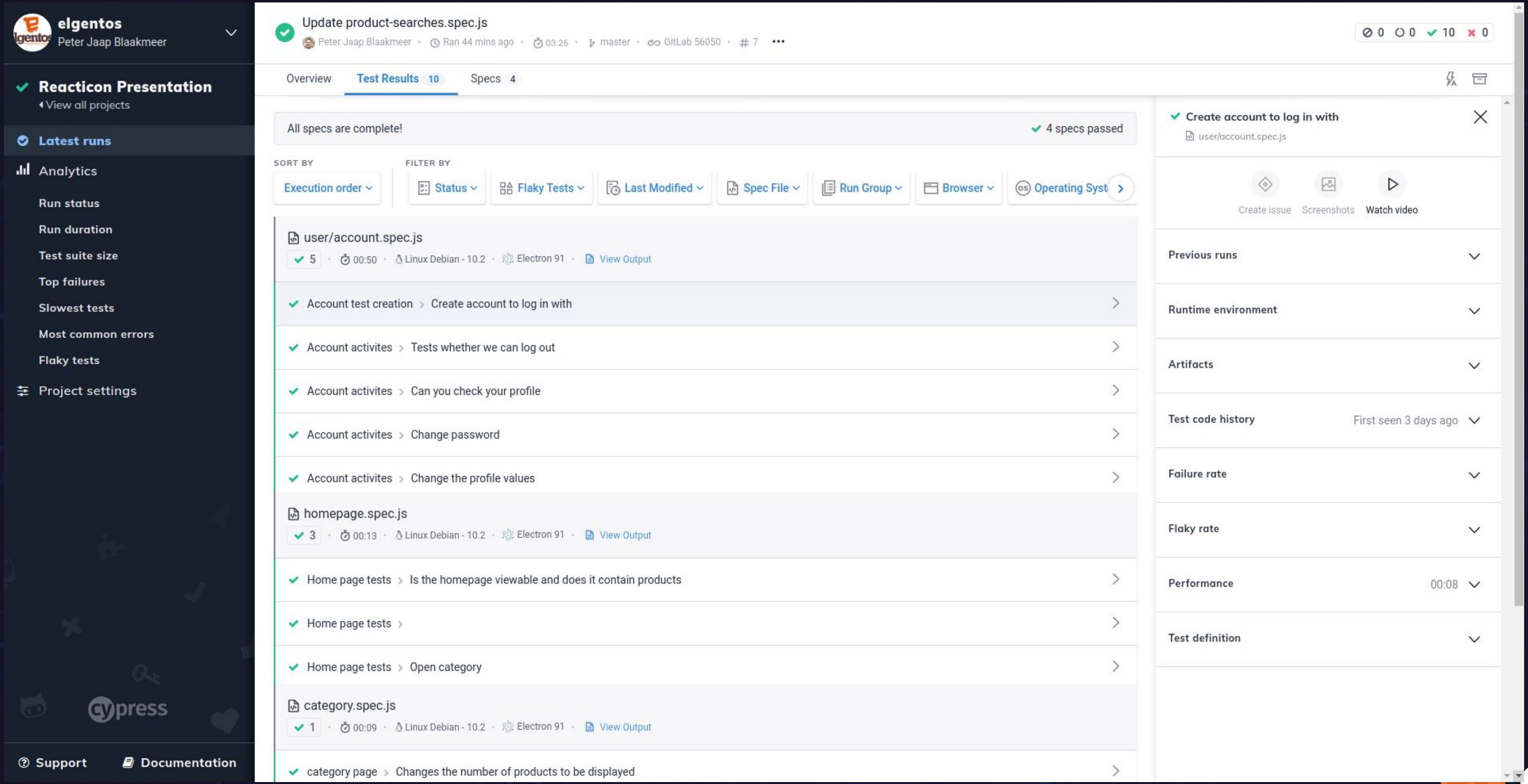
✓ No commit message available
Not available · Ran 2 months ago · 00:14 · Not available · Not available · # 3
⊘ 0 ○ 0 ✓ 5 ✗ 0

✗ Uncommented the PHP 7.4 setting
Jeroen Boersma · Ran 2 months ago · 00:12 · master · Not available · # 2
⊘ 0 ○ 0 ✓ 1 ✗ 1

✗ Uncommented the PHP 7.4 setting
Jeroen Boersma · Ran 2 months ago · 05:01 · master · Not available · # 1
⊘ 0 ○ 0 ✓ 119 ✗ 2

elgentos
ecommerce solutions

Cypress Dashboard



Update product-searches.spec.js

Peter Jaap Blaakmeer • Ran 44 mins ago • 03:26 • master • GitLab 56050 • # 7 •••

⊘ 0 ○ 0 ✓ 10 ✕ 0

All specs are complete! ✓ 4 specs passed

SORT BY: Execution order

FILTER BY: Status, Flaky Tests, Last Modified, Spec File, Run Group, Browser, Operating Syst

user/account.spec.js
✓ 5 • 00:50 • Linux Debian - 10.2 • Electron 91 • View Output

- ✓ Account test creation › Create account to log in with
- ✓ Account activites › Tests whether we can log out
- ✓ Account activites › Can you check your profile
- ✓ Account activites › Change password
- ✓ Account activites › Change the profile values

homepage.spec.js
✓ 3 • 00:13 • Linux Debian - 10.2 • Electron 91 • View Output

- ✓ Home page tests › Is the homepage viewable and does it contain products
- ✓ Home page tests ›
- ✓ Home page tests › Open category

category.spec.js
✓ 1 • 00:09 • Linux Debian - 10.2 • Electron 91 • View Output

- ✓ category page › Changes the number of products to be displayed

✓ Create account to log in with
user/account.spec.js

Previous runs

Runtime environment

Artifacts

Test code history: First seen 3 days ago

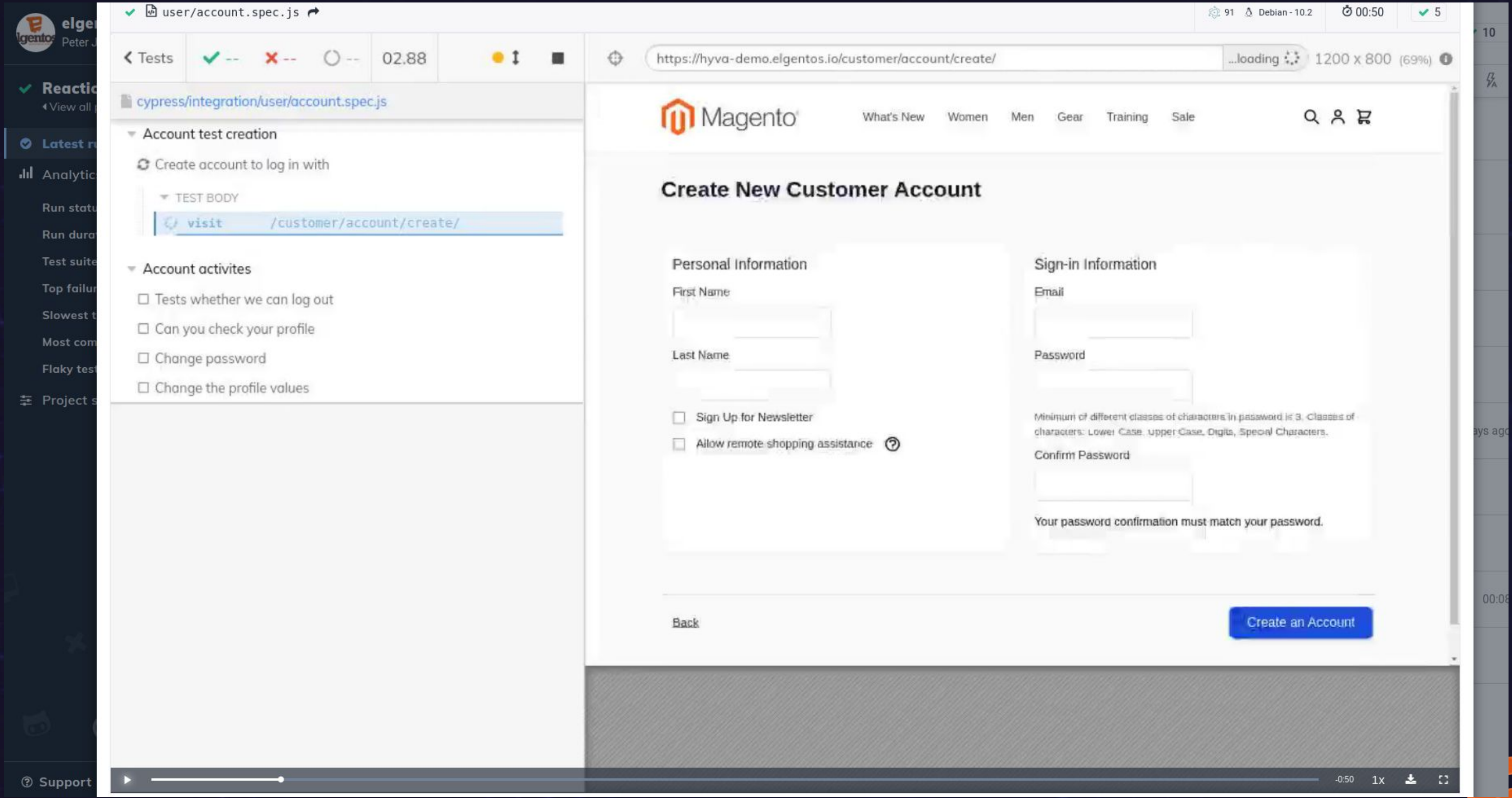
Failure rate

Flaky rate

Performance: 00:08

Test definition

Cypress Dashboard



✓ user/account.spec.js

91 Debian - 10.2 00:50 ✓ 5

< Tests ✓ -- ✗ -- ○ -- 02.88

https://hyva-demo.elgentos.io/customer/account/create/ ...loading 1200 x 800 (69%)

cypress/integration/user/account.spec.js

- Account test creation
  - Create account to log in with
    - TEST BODY
      - visit /customer/account/create/
- Account activites
  - □ Tests whether we can log out
  - □ Can you check your profile
  - □ Change password
  - □ Change the profile values

Create New Customer Account

Personal Information

First Name

Last Name

□ Sign Up for Newsletter

□ Allow remote shopping assistance

Sign-in Information

Email

Password

Minimum of different classes of characters in password is 3. Classes of characters: Lower Case, Upper Case, Digits, Special Characters.

Confirm Password

Your password confirmation must match your password.

Back

Create an Account

Cypress in an agency environment

Three ways how we use Cypress;

- Generic testing suite
- Specific regression tests
- Scheduled tests on production

Usage 1 - Generic testing suite

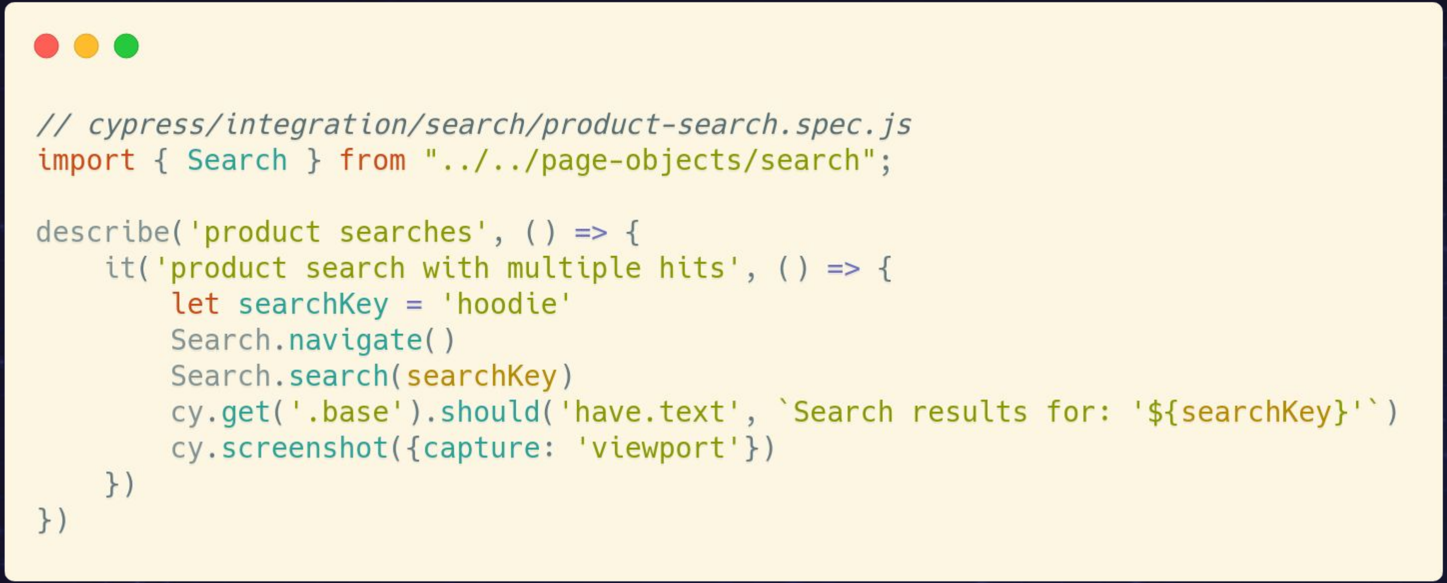
- Main goal; gain confidence nothing broke!
- You know immediately what broke
- Extremely useful after upgrades
- Copy into project & adjust where necessary
- Pre-deploy and post-deploy

Usage 1 - Generic testing suite

Tests the following areas in Magento:

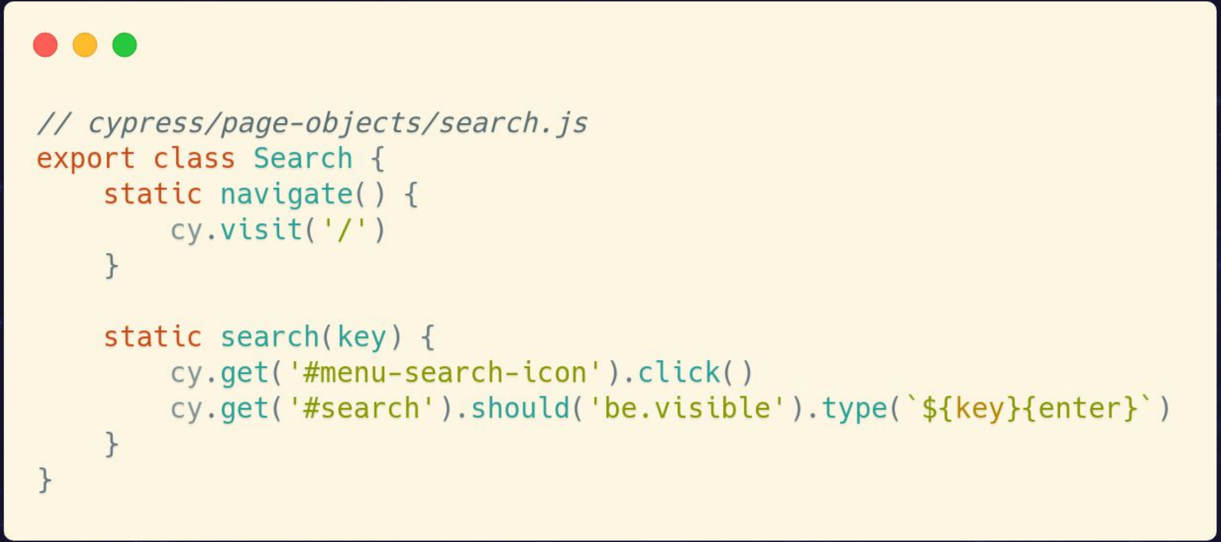
- Homepage
- Product page
- Category page
- Cart
- Checkout
- My Account
- Contact form
- Newsletter subscription

Usage 1 - Generic testing suite - example test



```
// cypress/integration/search/product-search.spec.js
import { Search } from "../../page-objects/search";

describe('product searches', () => {
    it('product search with multiple hits', () => {
        let searchKey = 'hoodie'
        Search.navigate()
        Search.search(searchKey)
        cy.get('.base').should('have.text', `Search results for: '${searchKey}'`)
        cy.screenshot({capture: 'viewport'})
    })
})
```

Usage 1 - Generic testing suite - example test

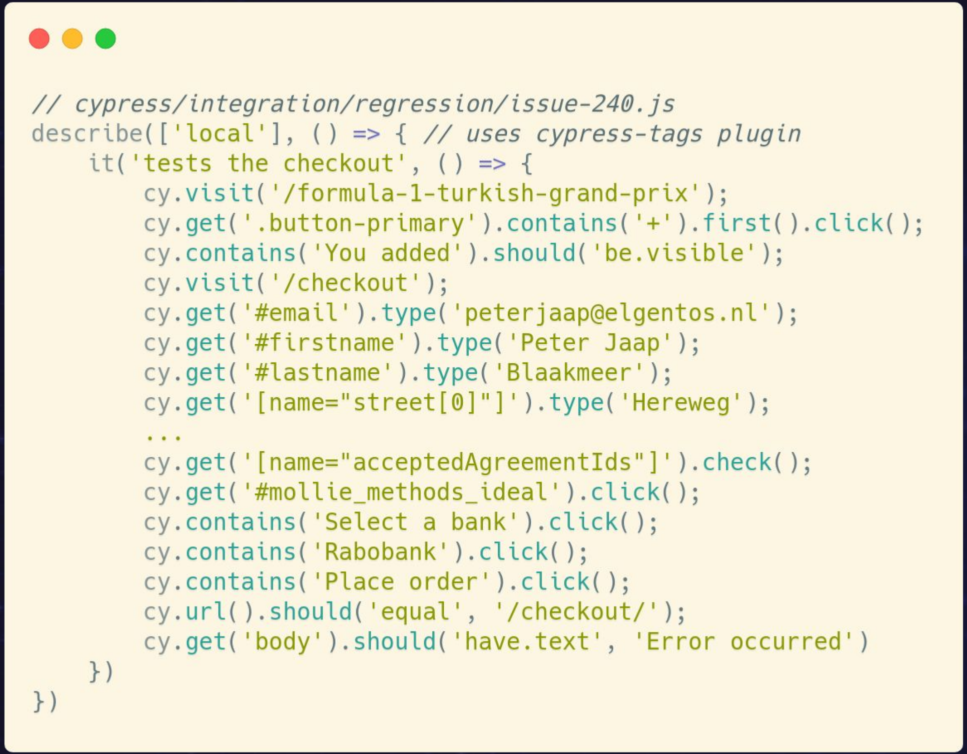


```
// cypress/page-objects/search.js
export class Search {
    static navigate() {
        cy.visit('/')
    }

    static search(key) {
        cy.get('#menu-search-icon').click()
        cy.get('#search').should('be.visible').type(`${key}{enter}`)
    }
}
```

Usage 2 - Specific regression tests

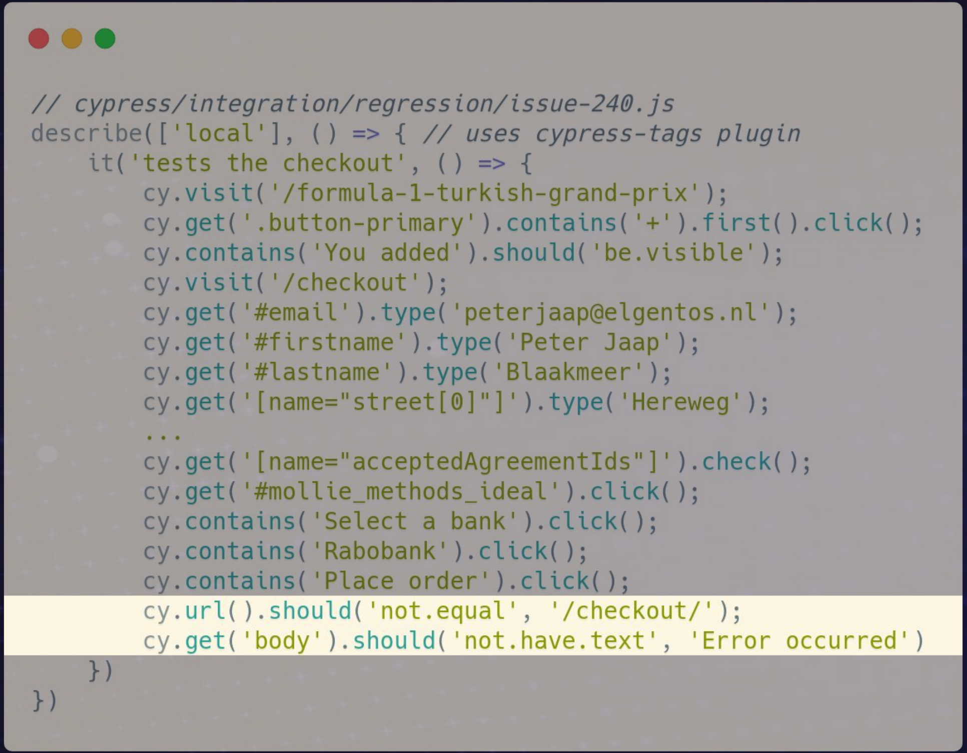
- Main goal: customer never sees a bug twice
- Bugfixing
  - Step 1; write a Cypress test to reproduce error
  - Step 2; fix bug
  - Step 3; invert assertion in test
- Company-wide policy
- Massive time-saver

Usage 2 - Specific regression tests - write reproduction test



```
// cypress/integration/regression/issue-240.js
describe(['local'], () => { // uses cypress-tags plugin
    it('tests the checkout', () => {
        cy.visit('/formula-1-turkish-grand-prix');
        cy.get('.button-primary').contains('+').first().click();
        cy.contains('You added').should('be.visible');
        cy.visit('/checkout');
        cy.get('#email').type('peterjaap@elgentos.nl');
        cy.get('#firstname').type('Peter Jaap');
        cy.get('#lastname').type('Blaakmeer');
        cy.get('[name="street[0]"]').type('Hereweg');
        ...
        cy.get('[name="acceptedAgreementIds"]').check();
        cy.get('#mollie_methods_ideal').click();
        cy.contains('Select a bank').click();
        cy.contains('Rabobank').click();
        cy.contains('Place order').click();
        cy.url().should('equal', '/checkout/');
        cy.get('body').should('have.text', 'Error occurred')
    })
})
```

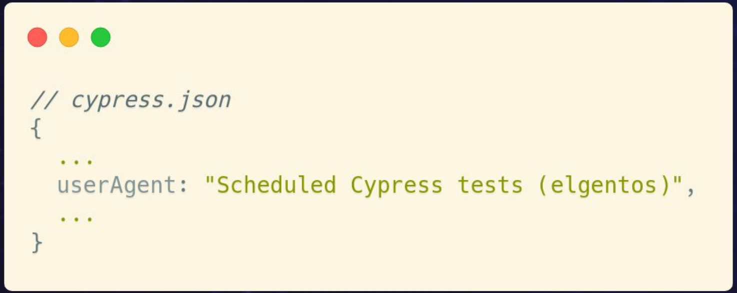
Usage 2 - Specific regression tests - invert test



```
// cypress/integration/regression/issue-240.js
describe(['local'], () => { // uses cypress-tags plugin
    it('tests the checkout', () => {
        cy.visit('/formula-1-turkish-grand-prix');
        cy.get('.button-primary').contains('+').first().click();
        cy.contains('You added').should('be.visible');
        cy.visit('/checkout');
        cy.get('#email').type('peterjaap@elgentos.nl');
        cy.get('#firstname').type('Peter Jaap');
        cy.get('#lastname').type('Blaakmeer');
        cy.get('[name="street[0]"]').type('Hereweg');
        ...
        cy.get('[name="acceptedAgreementIds"]').check();
        cy.get('#mollie_methods_ideal').click();
        cy.contains('Select a bank').click();
        cy.contains('Rabobank').click();
        cy.contains('Place order').click();
        cy.url().should('not.equal', '/checkout/');
        cy.get('body').should('not.have.text', 'Error occurred')
    })
})
```

Usage 3 - Scheduled tests on production

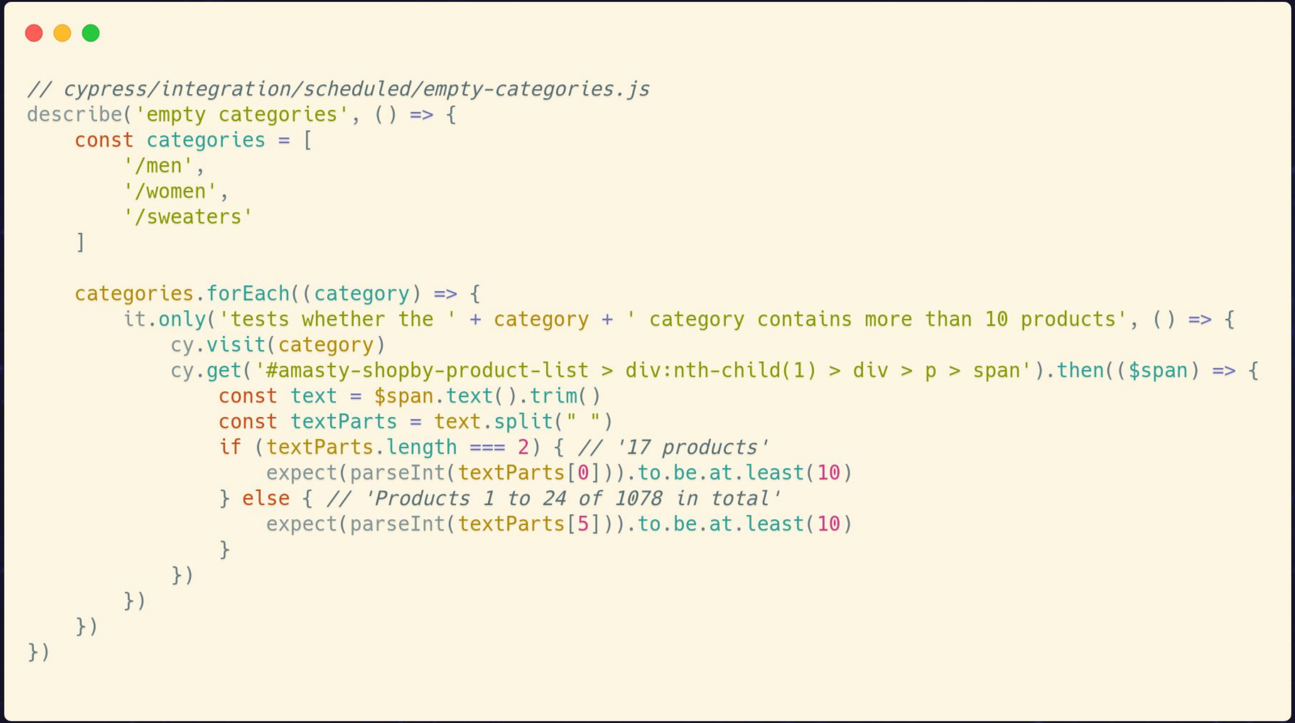
- Main goal: gather information / notifications
- We use Gitlab CI scheduled pipelines & Slack notifications
- Make 100% sure these are read-only tests
- Add a user-agent to identify the tests to exclude them in Google Analytics



```
// cypress.json
{
  ...
  userAgent: "Scheduled Cypress tests (elgentos)",
  ...
}
```

Usage 3 - Scheduled tests - example test

Tests whether certain categories still contain at least 10 products

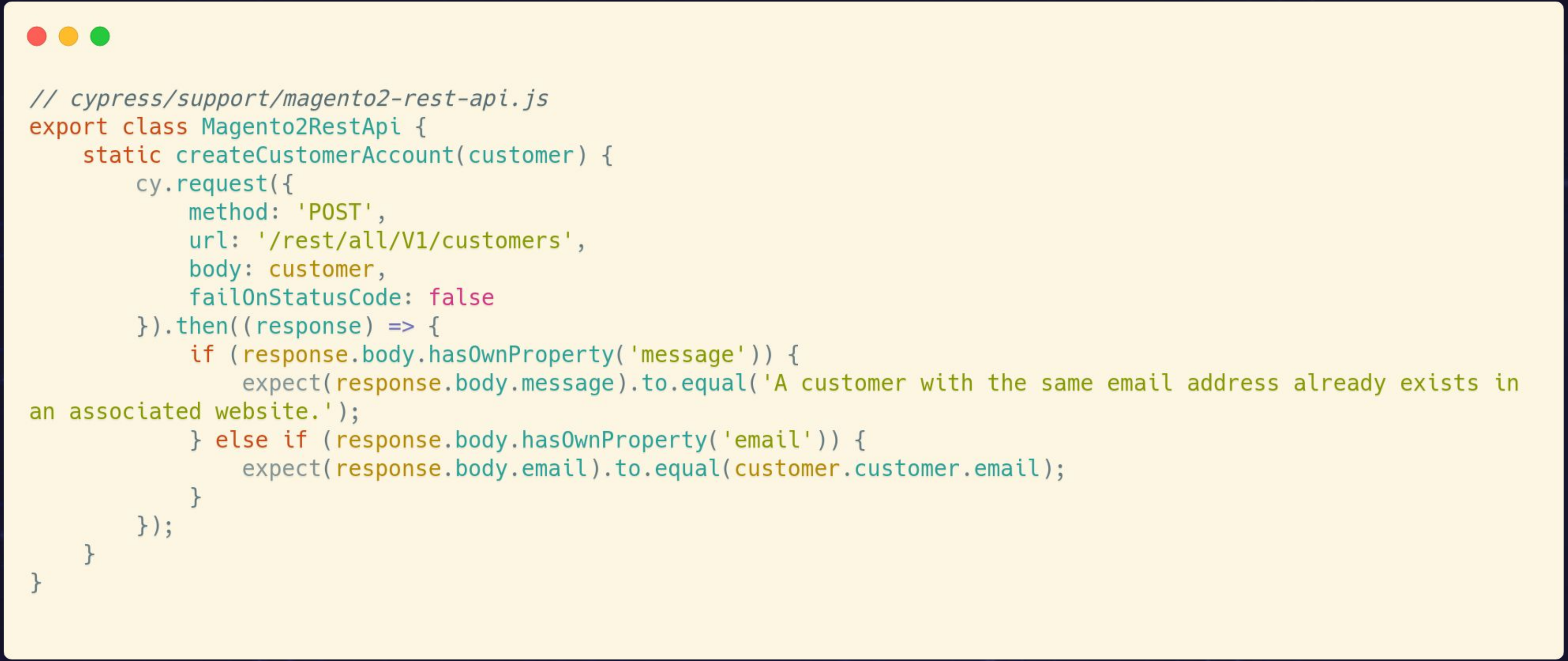


```
// cypress/integration/scheduled/empty-categories.js
describe('empty categories', () => {
    const categories = [
        '/men',
        '/women',
        '/sweaters'
    ]

    categories.forEach((category) => {
        it.only('tests whether the ' + category + ' category contains more than 10 products', () => {
            cy.visit(category)
            cy.get('#amasty-shopby-product-list > div:nth-child(1) > div > p > span').then(($span) => {
                const text = $span.text().trim()
                const textParts = text.split(" ")
                if (textParts.length === 2) { // '17 products'
                    expect(parseInt(textParts[0])).to.be.at.least(10)
                } else { // 'Products 1 to 24 of 1078 in total'
                    expect(parseInt(textParts[5])).to.be.at.least(10)
                }
            })
        })
    })
})
```

Magento-specific tricks

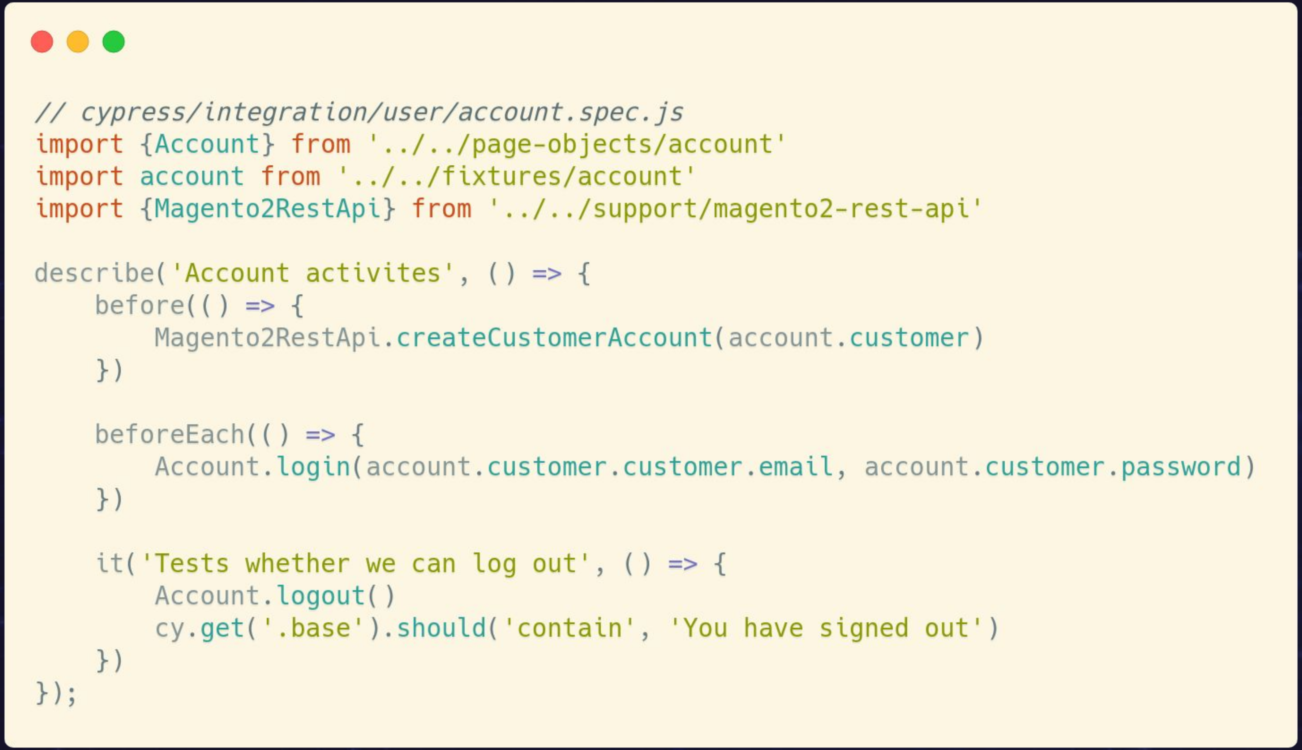
- Tag tests in groups for local, review & production
- We are experimenting with a Magento extension that makes fixtures like testable product URLs configurable
- Test the UI once, then use the REST API to speed up tests

Magento 2 REST API support class



```
// cypress/support/magento2-rest-api.js
export class Magento2RestApi {
    static createCustomerAccount(customer) {
        cy.request({
            method: 'POST',
            url: '/rest/all/V1/customers',
            body: customer,
            failOnStatusCode: false
        }).then((response) => {
            if (response.body.hasOwnProperty('message')) {
                expect(response.body.message).to.equal('A customer with the same email address already exists in
an associated website.');
            } else if (response.body.hasOwnProperty('email')) {
                expect(response.body.email).to.equal(customer.customer.email);
            }
        });
    }
}
```

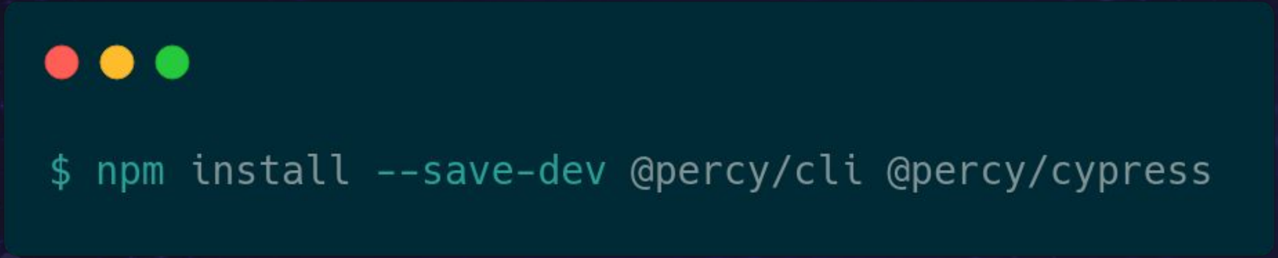
Accompanying test



```
// cypress/integration/user/account.spec.js
import {Account} from '../../page-objects/account'
import account from '../../fixtures/account'
import {Magento2RestApi} from '../../support/magento2-rest-api'

describe('Account activites', () => {
    before(() => {
        Magento2RestApi.createCustomerAccount(account.customer)
    })

    beforeEach(() => {
        Account.login(account.customer.customer.email, account.customer.password)
    })

    it('Tests whether we can log out', () => {
        Account.logout()
        cy.get('.base').should('contain', 'You have signed out')
    })
});
```

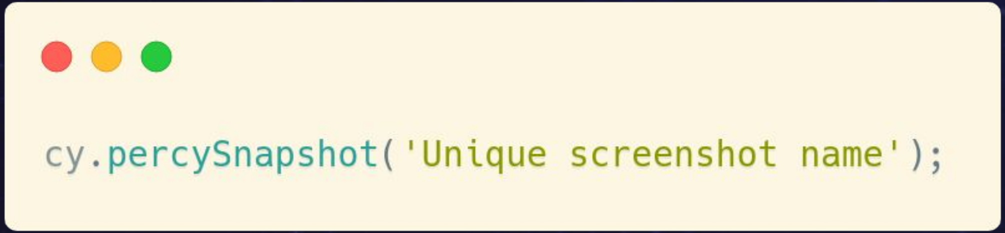
Visual testing

- Percy.io works in perfect tandem with Cypress
- SaaS offering with a dashboard
- Easy to install



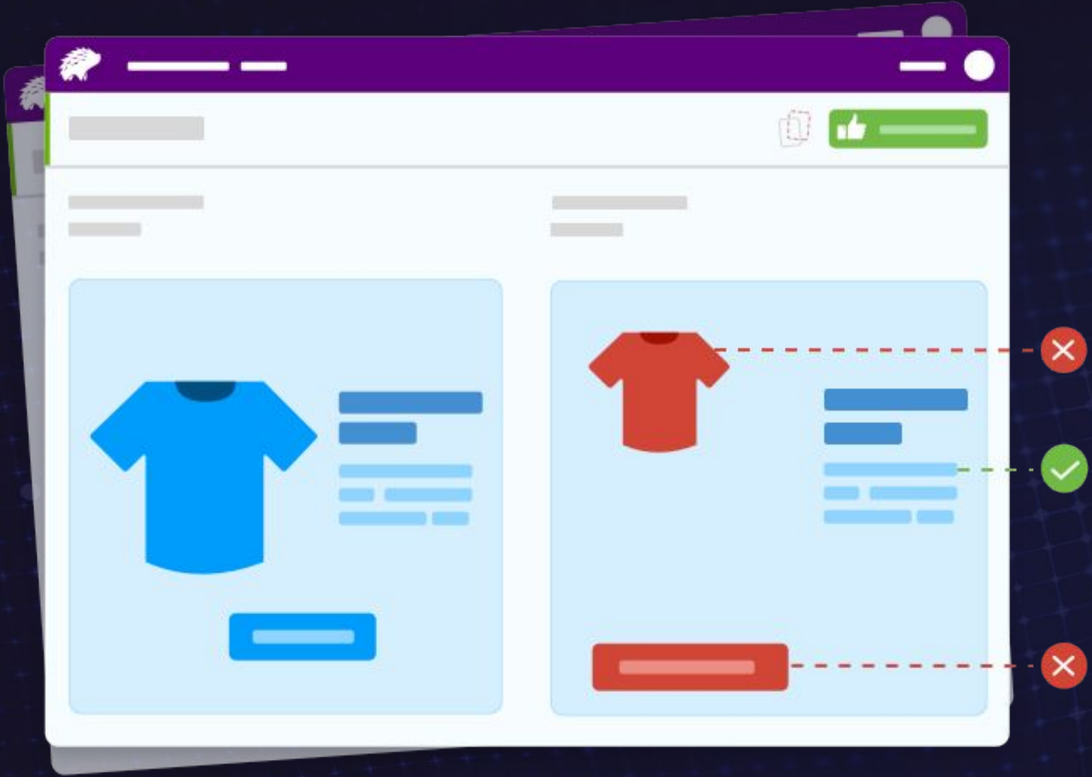
```
$ npm install --save-dev @percy/cli @percy/cypress
```

- Easy to use



```
cy.percySnapshot('Unique screenshot name');
```

Visual testing



- Visual diffs
- Configurable notification threshold

Some alternatives to Percy:

- Applitools Eyes
- Happo.io
- Micoo (open source)

Plugins

- The previously shown Percy package is actually a Cypress plugin
- There are over a 100 plugins available
- Some useful plugins I use:

cypress-tags	tag tests to group them into local, review, production
cypress-file-upload	add file upload support
cypress-drag-drop	add drag & drop support
cypress-plugin-tab	{tab}, like {enter}
cypress-commands	extends base commands and adds a few
cypress-mailosaur	to test outgoing mail
cy-view	run specific tests against different viewports

Thanks for watching!

Stay tuned:

- We will **open-source** the generic testing suite for Magento 2 - sponsored by elgentos

- I am creating a **video course** on Cypress testing for e-commerce - planning to launch Q1 2022

elgentos
ecommerce solutions