versus

Yann Helleboid

Orange Innovation 2023

Cypress, Playwright, what ?

Test automation tools

RPA capabilities

Cypress, Playwright, why ?

Open source / Free

Easy / Powerful

Cypress, Playwright, what for ?

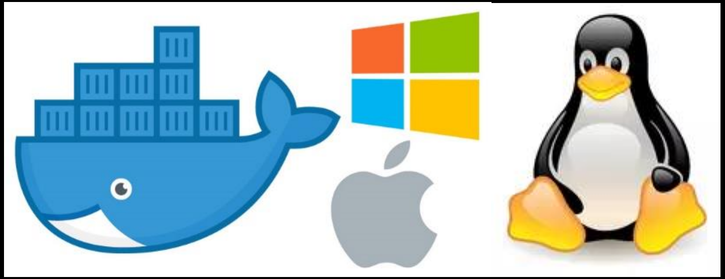
HTTP oriented functional tests

GUI and API

Chrome Firefox Edge Electron Brave Webkit / Safari

Cypress, Playwright, where ?

Most platforms



Your machine or on a VM or on a cloud

Cypress, Playwright, how ?

Hands-on training

see by yourselves

means it’s your turn to type

Let’s go !

Cypress, Playwright, what is different ?

friendly ui
better integrated js/browser
standard libs included

step-by-step debug possible
js + python + others
browser tab management

exercise 1

goal : install the software

install the environment

install a proper editor or IDE

notepad++, eclipse, vscode, vim …

install node.js

Go to https://nodejs.org/en

Follow the install guide for your OS

configure for Orange proxies

eburo => dev access VPN
yourdev => set proxy.rd.francetelecom.fr

* For docker, start here : https://docs.cypress.io/examples/examples/docker

launch : 2 modes

open in a GUI => design, debug and analyse
    use headed browser

run as a CLI => run tests
    use headless browser (or headed)

Cypress, Playwright and browsers

Cypress is shipped with a electron browser

Cypress can use installed standard browsers

Cypress can control headed or headless browsers

Playwright is shipped with out any browser

You must install specific browsers for Playwright

Playwright can control headed or headless browsers

install

exercise 1

```
λ mkdir cypress_training

λ cd cypress_training

λ npm init -y

λ npm install cypress
```

```
λ mkdir playwright_training

λ cd playwright_training

λ npm init playwright
```

```
Getting started with writing end-to-end tests with Playwright:
Initializing project in '.'
√ Do you want to use TypeScript or JavaScript? · JavaScript
√ Where to put your end-to-end tests? · tests
√ Add a GitHub Actions workflow? (y/N) · false
√ Install Playwright browsers (can be done manually via 'npx
playwright install')? (Y/n) · true
```

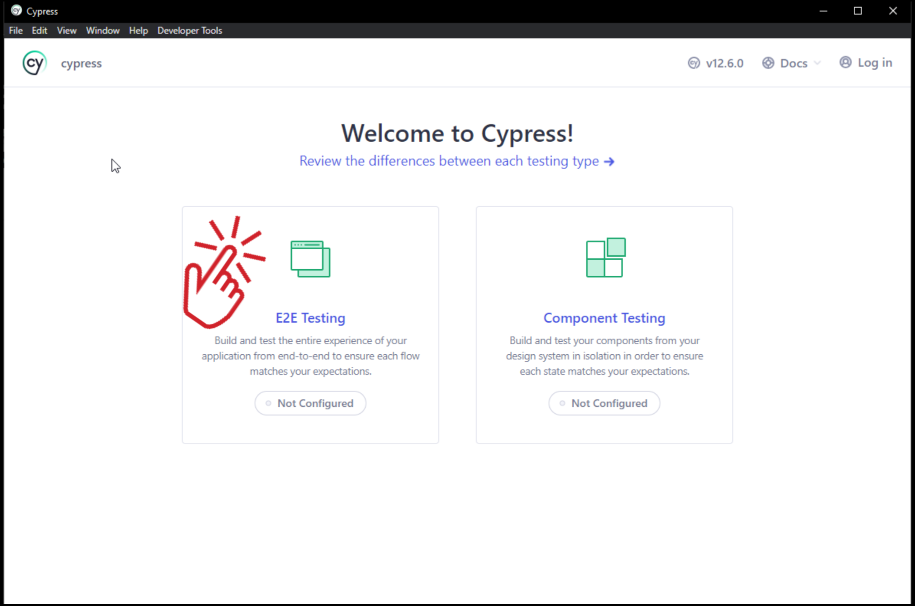
install

exercise 1

```
λ npx cypress open
```

```
λ npx playwright test --ui
λ npx playwright test --debug
```

install



Welcome to Cypress!

Review the differences between each testing type →

E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

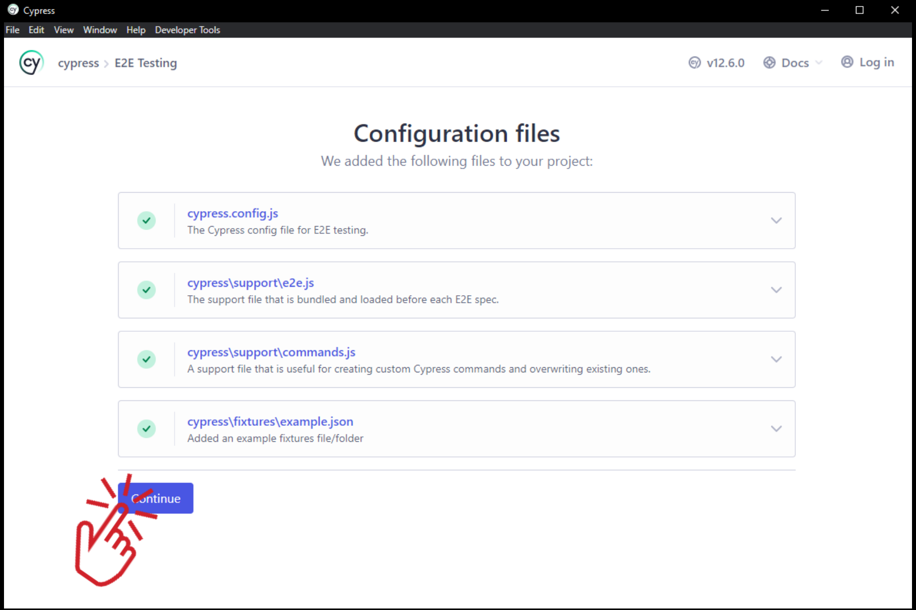
Not Configured

Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

Not Configured

install



Configuration files

We added the following files to your project:

cypress.config.js
The Cypress config file for E2E testing.

cypress\support\e2e.js
The support file that is bundled and loaded before each E2E spec.

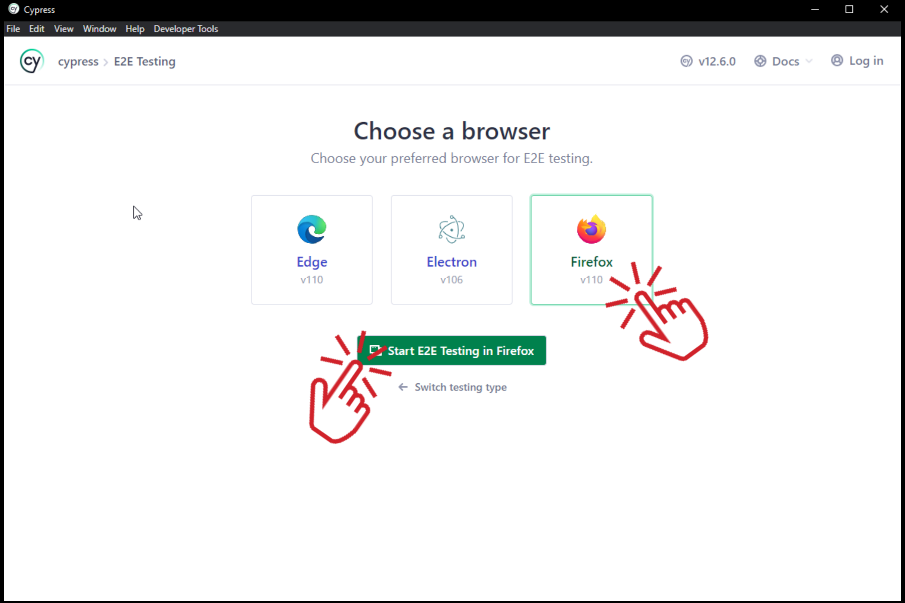
cypress\support\commands.js
A support file that is useful for creating custom Cypress commands and overwriting existing ones.

cypress\fixtures\example.json
Added an example fixtures file/folder

Continue

exercise 1

install

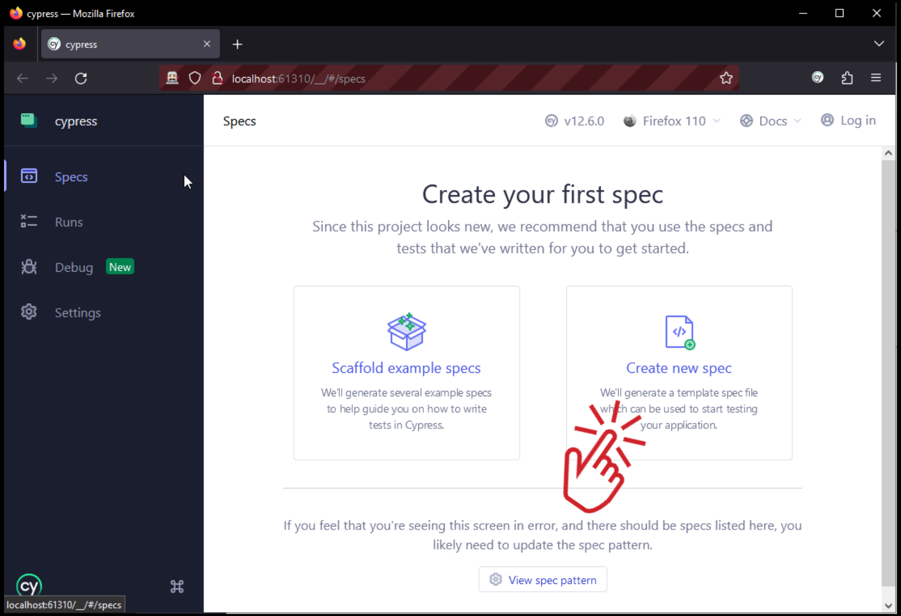


Choose a browser

Choose your preferred browser for E2E testing.

install

exercise 1



cypress — Mozilla Firefox

localhost:61310/__/#/specs

Create your first spec

Since this project looks new, we recommend that you use the specs and tests that we've written for you to get started.

Scaffold example specs

We'll generate several example specs to help guide you on how to write tests in Cypress.

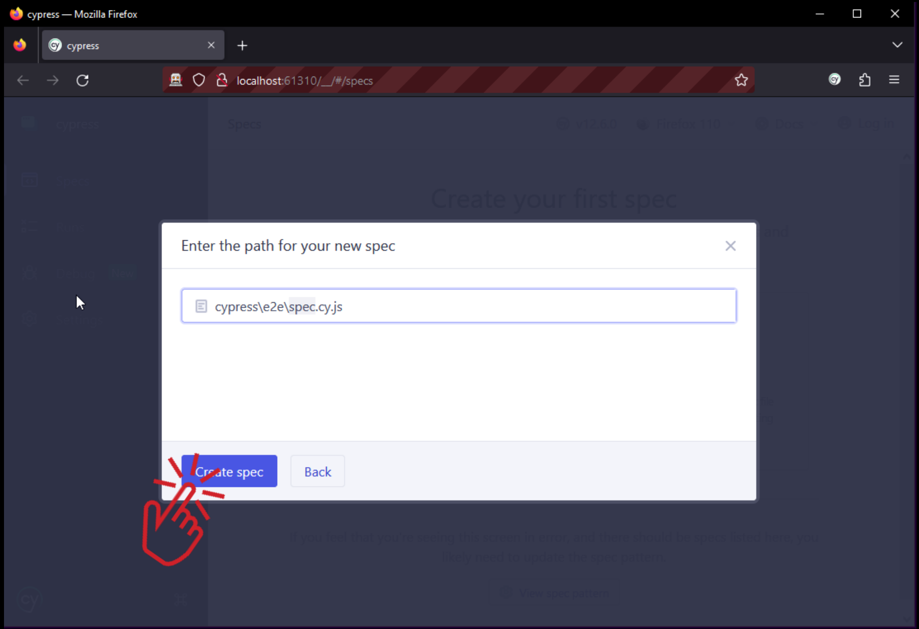
Create new spec

We'll generate a template spec file which can be used to start testing your application.

If you feel that you're seeing this screen in error, and there should be specs listed here, you likely need to update the spec pattern.

View spec pattern

install

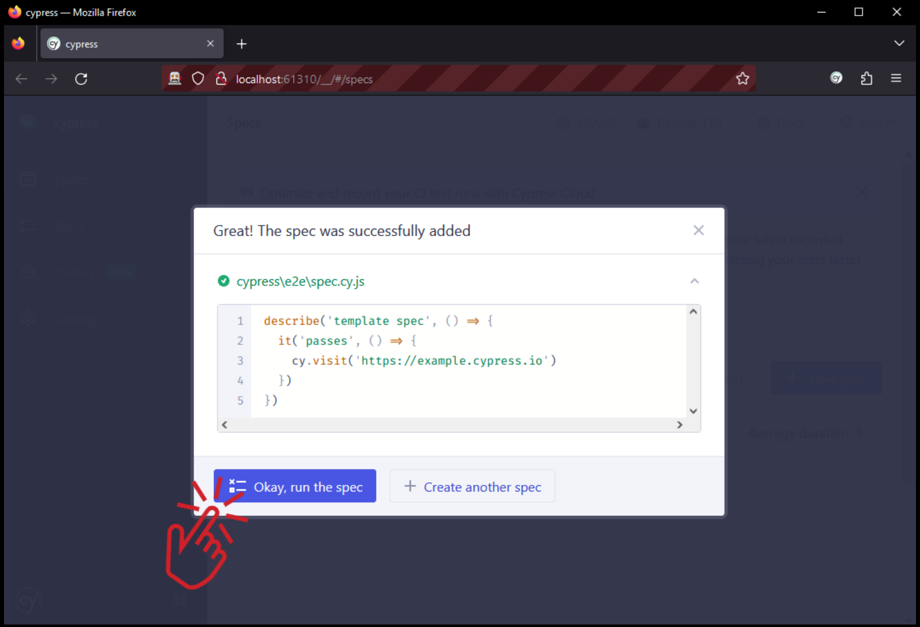

cypress — Mozilla Firefox

localhost:61310/__/#/specs

Enter the path for your new spec

```
cypress\e2e\spec.cy.js
```

install

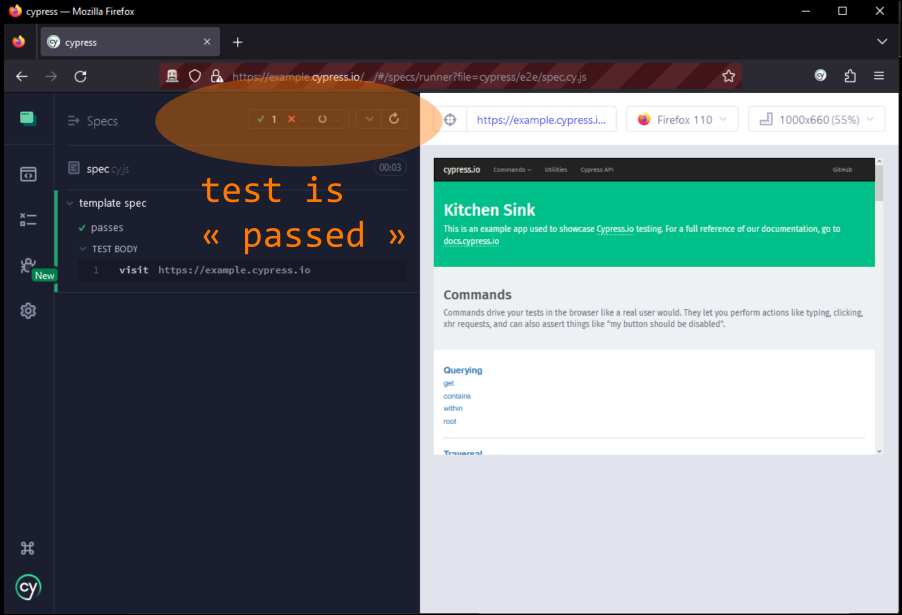


Great! The spec was successfully added

cypress\e2e\spec.cy.js

```
describe('template spec', () => {
  it('passes', () => {
    cy.visit('https://example.cypress.io')
  })
})
```

install



cypress — Mozilla Firefox

https://example.cypress.io/__/#/specs/runner?file=cypress/e2e/spec.cy.js

Specs ✓ 1 ✕ -- ○ --

https://example.cypress.i... Firefox 110 1000x660 (55%)

spec.cy.js 00:03

template spec

✓ passes

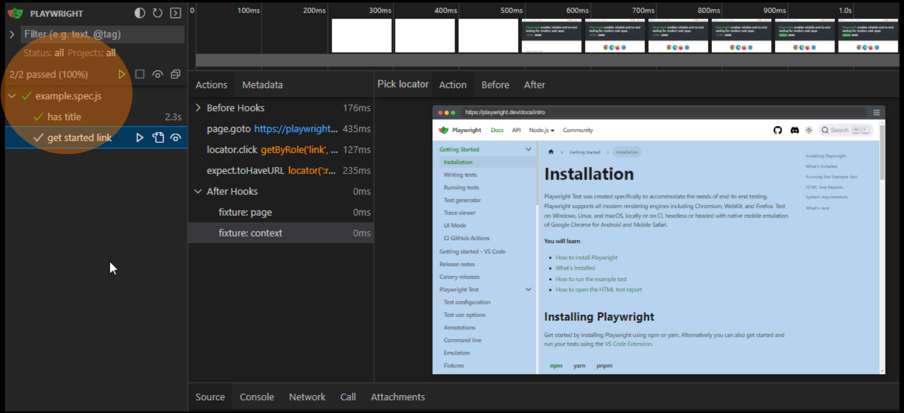
TEST BODY

```
visit https://example.cypress.io
```

test is « passed »

cypress.io
Commands
Utilities
Cypress API
GitHub
Kitchen Sink
This is an example app used to showcase Cypress.io testing. For a full reference of our documentation, go to docs.cypress.io
Commands
Commands drive your tests in the browser like a real user would. They let you perform actions like typing, clicking, xhr requests, and can also assert things like "my button should be disabled".
Querying
get
contains
within
root
New

exercise 1



PLAYWRIGHT

Filter (e.g. text, @tag)

Status: all Projects: all

2/2 passed (100%)

✓ example.spec.js

✓ has title 2.3s

✓ get started link

Actions	
› Before Hooks	176ms
page.goto https://playwright...	435ms
locator.click getByRole('link', ...	127ms
expect.toHaveURL locator(':r...	235ms
⌄ After Hooks	0ms
fixture: page	0ms
fixture: context	0ms

run headless

headless run will be used in a ci pipeline

exercise 2

goal : headless run

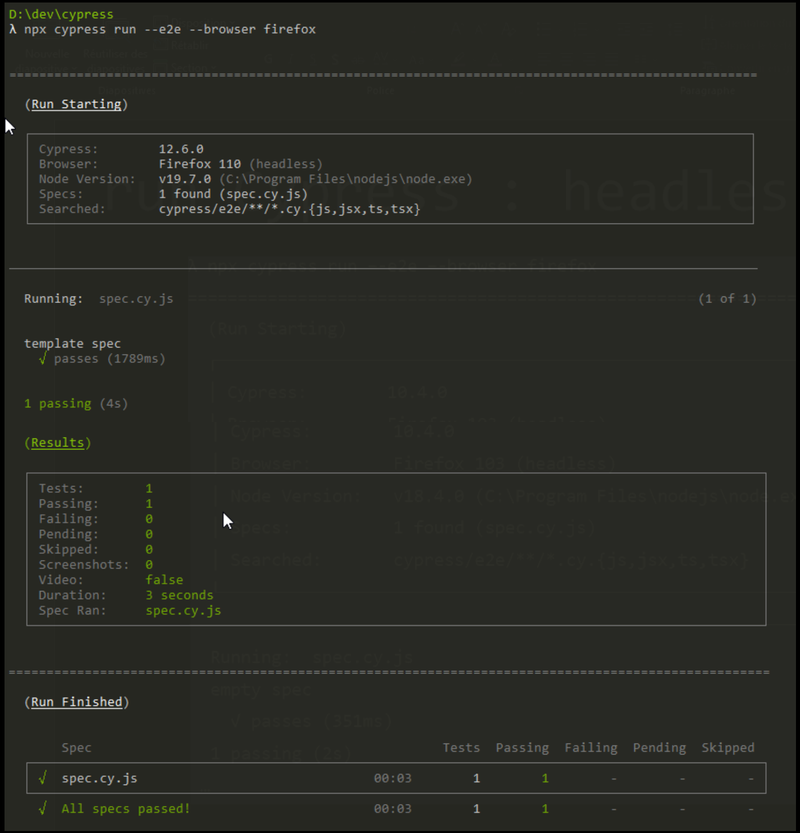
run headless

exercise 2

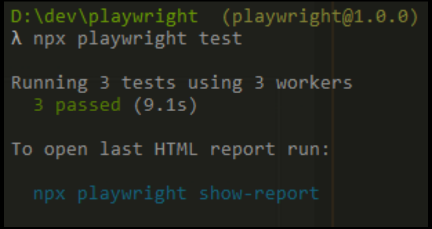
```
λ npx cypress run --e2e
--browser firefox
```

```
λ npx playwright test
--project=firefox
```

run headless



```
D:\dev\cypress
λ npx cypress run --e2e --browser firefox

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

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:        12.6.0                                                                         │
  │ Browser:        Firefox 110 (headless)                                                         │
  │ Node Version:   v19.7.0 (C:\Program Files\nodejs\node.exe)                                     │
  │ Specs:          1 found (spec.cy.js)                                                           │
  │ Searched:       cypress/e2e/**/*.cy.{js,jsx,ts,tsx}                                            │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘


────────────────────────────────────────────────────────────────────────────────────────────────────

  Running:  spec.cy.js                                                                      (1 of 1)


  template spec
    √ passes (1789ms)


  1 passing (4s)


  (Results)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Tests:        1                                                                                │
  │ Passing:      1                                                                                │
  │ Failing:      0                                                                                │
  │ Pending:      0                                                                                │
  │ Skipped:      0                                                                                │
  │ Screenshots:  0                                                                                │
  │ Video:        false                                                                            │
  │ Duration:     3 seconds                                                                        │
  │ Spec Ran:     spec.cy.js                                                                       │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘


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

  (Run Finished)


       Spec                                              Tests  Passing  Failing  Pending  Skipped
  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ √  spec.cy.js                               00:03        1        1        -        -        - │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘
    √  All specs passed!                        00:03        1        1        -        -        -
```



```
D:\dev\playwright  (playwright@1.0.0)
λ npx playwright test

Running 3 tests using 3 workers
  3 passed (9.1s)

To open last HTML report run:

  npx playwright show-report
```

exercise 2

configuration

Cypress is shipped with a default config :

cypress.config.js

Playwright is shipped with a default config :

playwright.config.js

exercise 3

goal : configure the tools

configuration

exercise 3

```
λ cat cypress.config.js
const { defineConfig } = require("cypress");
module.exports = defineConfig(
{
  video: false,
  e2e:
  {
    baseUrl:"https://ozh.github.io/cookieclicker",
    setupNodeEvents(on, config) {}
  }
});
```

control video creation
(run mode only)

default url

```
λ cat playwright.config.js
const { defineConfig, devices } =
require('@playwright/test');
module.exports = defineConfig({
/* Run tests in files in parallel */
  fullyParallel: false,
use: { actionTimeout: 0,
       video: 'off',
       /* Base URL to use in actions like
          `await    page.goto('/')`. */
       baseURL: 'https://ozh.github.io',
}
  /* Configure projects for major browsers */
  projects: [
    {  name: 'chromium',
       use: { ...devices['Desktop Chrome'] } },
    {  name: 'firefox',
       use: { ...devices['Desktop Firefox'] }
    {  name: 'webkit',
       use: { ...devices['Desktop Safari'] }
  ]
});
```

run test in //
CAUTION !!!

environment variables

Cypress can use environment variables from

- .env file (using dotenv)
- cypress.config.js
- cypress.env.json
- CYPRESS_* system variables
- cypress cli

Playwright can use environment variables from

- .env file (using dotenv)
- system variables

exercise 4

goal : use environment variables

environment variables

exercise 4

```
λ cat cypress.env.json
{
  "url": "https://ozh.github.io/cookieclicker"
}
```

create variable «url»

```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  it('test', () =>
  {
    cy.visit(Cypress.env('url'));
  })
})
```

use variable «url»

```
λ npm install dotenv
λ cat playwright.config.js
require('dotenv').config();
λ cat .env
url="https://ozh.github.io/cookieclicker"

λ cat tests/example.spec.js
const {test, expect} = require('@playwright/test');
test.describe('test suite', () =>
{
  test('test', async ({ page }) =>
  {
    await page.goto(process.env.url);
  });
});
```

log

log in Cypress

```
cy.log('my trace');
```

log in the browser console

```
console.log('my trace');
```

log in the browser console

```
await page.evaluate(() =>
console.log('my trace'));
```

exercise 5

goal : log debug traces

log

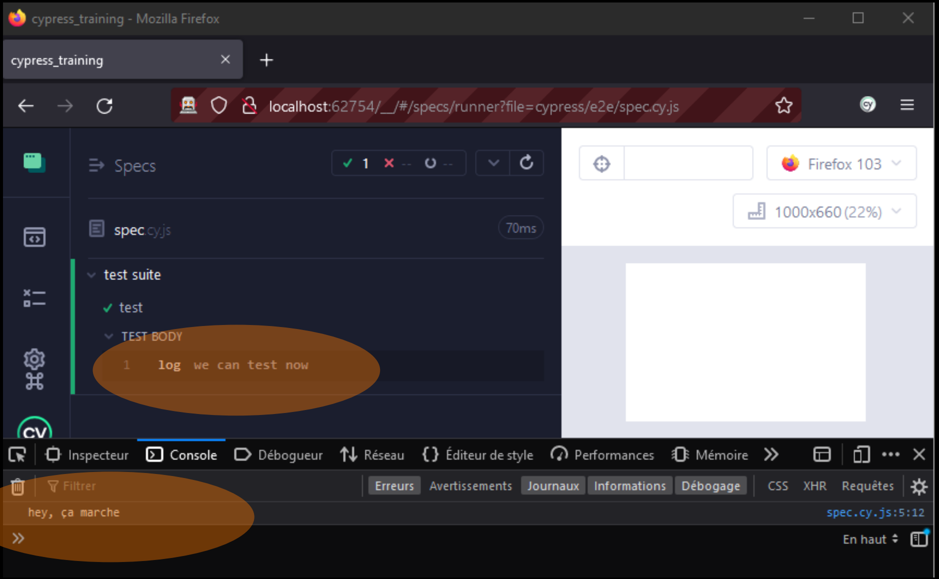
exercise 5

```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  it('test', () =>
  {
    console.log('hey, its working');
    cy.log('we can test now');
  });
});
```

```
λ cat tests/example.spec.js
const {test, expect} = require('@playwright/test');
test.describe('test suite', () =>
{
  test('test', async ({ page }) =>
  {
    await page.evaluate(()=> console.log('hello'));
    await page.waitForTimeout(20000);
  });
});
```

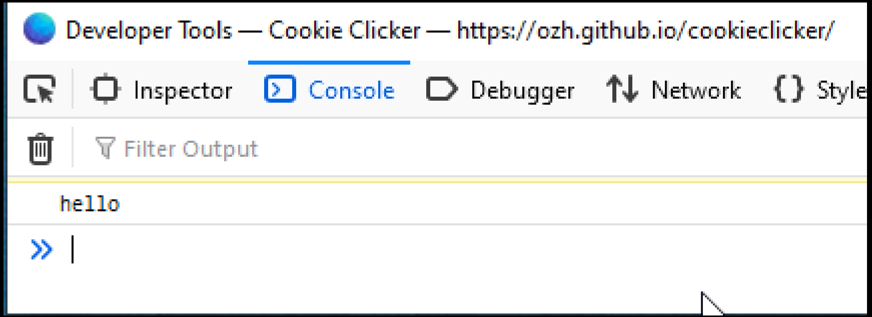
log

exercise 5



```
log  we can test now
```

```
hey, ça marche                                    spec.cy.js:5:12
```



Developer Tools — Cookie Clicker — https://ozh.github.io/cookieclicker/

```
hello
```

only seen in Playwright debug mode

load an url

load a url (web site or api route) with the browser

```
cy.visit('https://site.com');
    or
cy.visit('/');
```

```
await page.goto('https://site.com');
    or
await page.goto('/');
```

use default «baseUrl» defined in config

exercise 6

goal : load an url

load an url

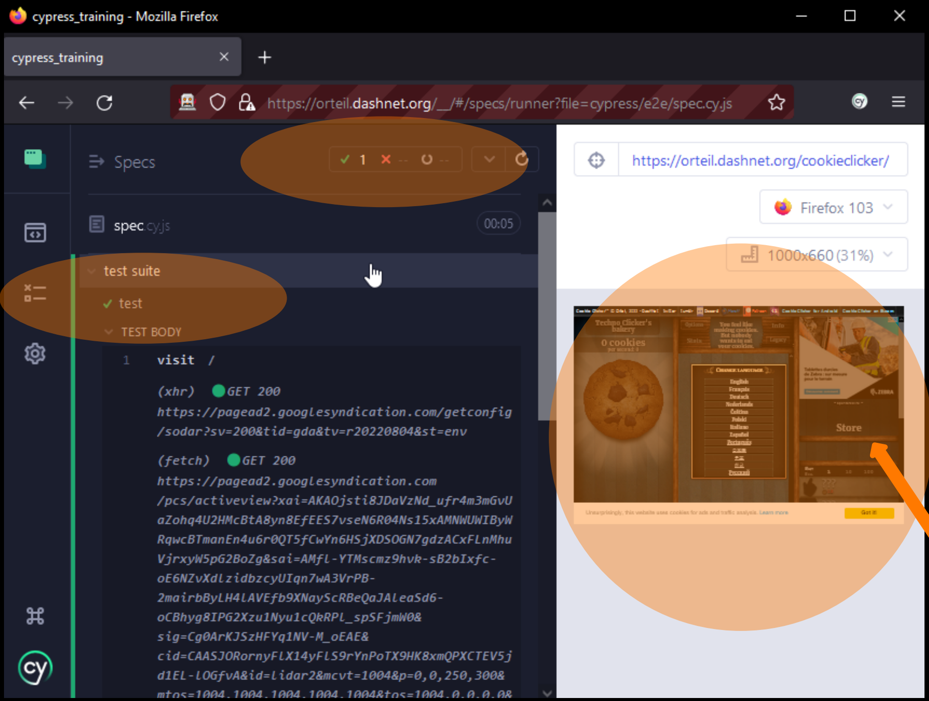
exercise 6

```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  it('test', () =>
  {
    cy.visit('/');
  });
});
```

```
λ cat tests/example.spec.js
const { test, expect } =
require('@playwright/test');
test.describe('test suite', () =>
{
  test('test', async ({page}) =>
  {
    await page.goto('/cookieclicker');
  });
});
```

load an url

exercise 6



cypress_training - Mozilla Firefox

https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

Specs ✓ 1 ✕ -- ○ --

spec.cy.js 00:05

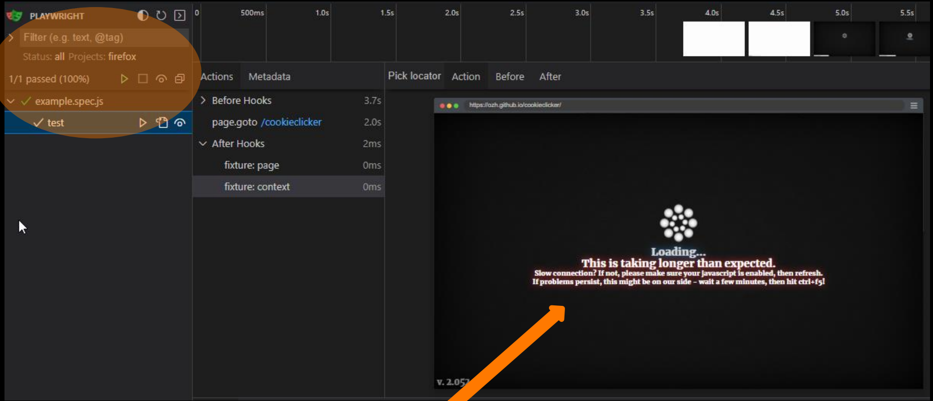
test suite
✓ test
TEST BODY

```
1  visit  /
   (xhr)  ● GET 200
   https://pagead2.googlesyndication.com/getconfig
   /sodar?sv=200&tid=gda&tv=r20220804&st=env
   (fetch)  ● GET 200
   https://pagead2.googlesyndication.com
   /pcs/activeview?xai=AKAOjsti8JDaVzNd_ufr4m3mGvU
   aZohq4U2HMcBtA8yn8EfEES7vseN6R04Ns15xAMNWUWIByW
   RqwcBTmanEn4u6r0QT5fCwYn6HSjXDSOGN7gdzACxFLnMhu
   VjrxyW5pG2BoZg&sai=AMfl-YTMscmz9hvk-sB2bIxfc-
   oE6NZvXdlzidbzcyUIqn7wA3VrPB-
   2mairbByLH4LAVEfb9XNayScRBeQaJALeaSd6-
   oCBhyg8IPG2Xzu1Nyu1cQkRPL_spSFjmW0&
   sig=Cg0ArKJSzHFYq1NV-M_oEAE&
   cid=CAASJORornyFLX14yFLS9rYnPoTX9HK8xmQPXCTEV5j
   d1EL-LOGfvA&id=lidar2&mcvt=1004&p=0,0,250,300&
   mtos=1004,1004,1004,1004,1004&tos=1004,0,0,0,0&
```

https://orteil.dashnet.org/cookieclicker/

Firefox 103

1000x660 (31%)



PLAYWRIGHT

Filter (e.g. text, @tag)
Status: all Projects: firefox
1/1 passed (100%)
✓ example.spec.js
✓ test

Actions	
Before Hooks	3.7s
page.goto /cookieclicker	2.0s
After Hooks	2ms
fixture: page	0ms
fixture: context	0ms

https://ozh.github.io/cookieclicker/

Loading...
This is taking longer than expected.
Slow connection? If not, please make sure your javascript is enabled, then refresh.
If problems persist, this might be on our side – wait a few minutes, then hit ctrl+f5!

note the difference
Cypress ends the test when everything is loaded
Playwright ends the test when the « loaded » event is fired

get, click and auto retry

get a DOM element using a selector
retrying (no wait needed)

```
cy.get(selector);
```

```
await page.locator(selector);
```

click on an actionable element
retrying (no wait needed)

```
cy.get(selector).click();
```

```
await page.locator(selector).click(
```

exercise 7

goal : select a DOM element and click on it

get, click and auto retry

exercise 7

```
λ cat cypress.env.json
{
  "lang":"#langSelect-EN",
  "care":".cc_btn_accept_all"
}
```

```
λ cat .env

  lang2="#langSelect-EN"
  care=".cc_btn_accept_all"
```

CAUTION : dotenv does not override existing varenv lang is existing so use another name

use specific test ids whenever it is possible
and encourage developers to add them or add them yourself

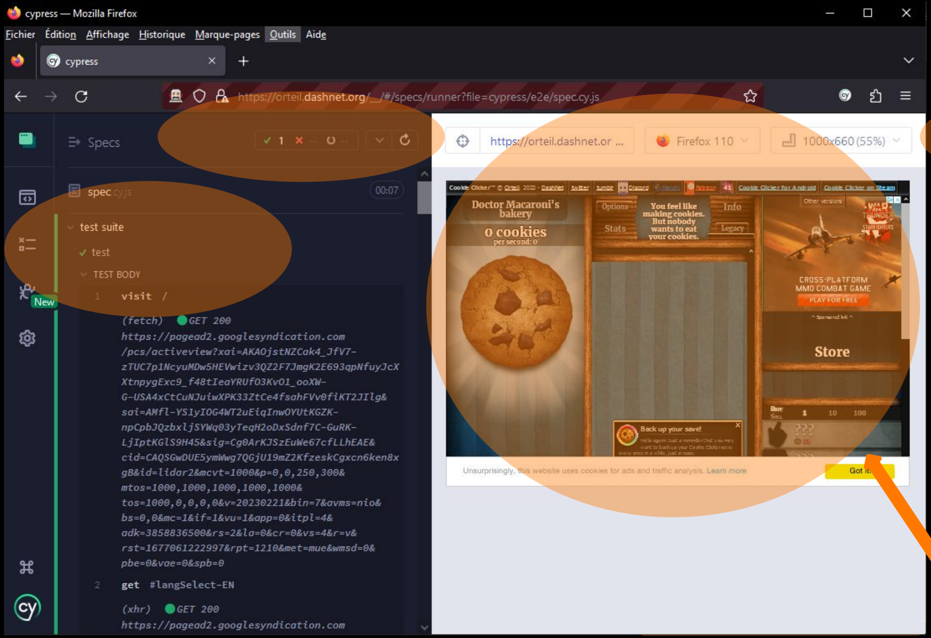
```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  …
  cy.get(Cypress.env('lang')).click();
  cy.get(Cypress.env('care')).click();
  …
});
```

```
λ cat tests/example.spec.js
test.describe('test suite', () =>
{
  …
  await page.locator(process.env.lang2).click();
  await page.locator(process.env.care).click();
  …
});
```

Re

get, click and auto retry

exercise 7



cypress — Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils Aide

https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

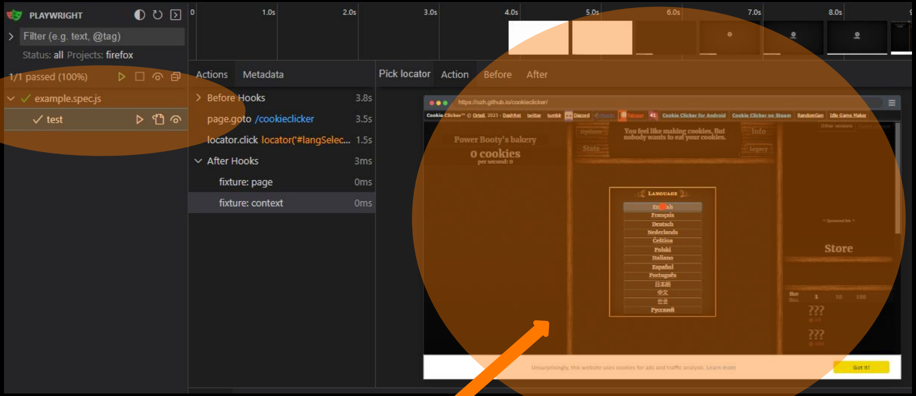
Specs ✓ 1 ✕ -- ○ --

spec.cy.js 00:07

test suite
✓ test
TEST BODY

```
visit  /
(fetch)  GET 200
https://pagead2.googlesyndication.com
/pcs/activeview?xai=AKAOjstNZCak4_JfV7-
zTUC7p1NcyuMDw5HEVwizv3QZ2F7JmgK2E693qpNfuyJcX
XtnpygExc9_f48tIeaYRUf03KvO1_ooXW-
G-USA4xCtCuNJuiwXPK33ZtCe4fsahFVv0fiKT2JIlg&
sai=AMfl-YS1yIOG4WT2uEiqInwOYUtKGZK-
npCpbJQzbxljSYWq03yTeqH2oDxSdnf7C-GuRK-
LjIptKGlS9H45&sig=Cg0ArKJSzEuWe67cfLLhEAE&
cid=CAQSGwDUE5ymWwg7QGjU19mZ2KfzeskCgxcn6ken8x
gB&id=lidar2&mcvt=1000&p=0,0,250,300&
mtos=1000,1000,1000,1000,1000&
tos=1000,0,0,0,0&v=20230221&bin=7&avms=nio&
bs=0,0&mc=1&if=1&vu=1&app=0&itpl=4&
adk=3858836500&rs=2&la=0&cr=0&vs=4&r=v&
rst=1677061222997&rpt=1210&met=mue&wmsd=0&
pbe=0&vae=0&spb=0
get  #langSelect-EN
(xhr)  GET 200
https://pagead2.googlesyndication.com
```

https://orteil.dashnet.or ... Firefox 110 1000x660 (55%)



PLAYWRIGHT

Filter (e.g. text, @tag)

Status: all Projects: firefox

1/1 passed (100%)

✓ example.spec.js
✓ test

Actions Metadata

Before Hooks	3.8s
page.goto /cookieclicker	3.5s
locator.click locator('#langSelec...	1.5s
After Hooks	3ms
fixture: page	0ms
fixture: context	0ms

Pick locator Action Before After

again note the difference
Cypress ends the test when
everything is loaded
Playwright ends the test
immediately

asserting

asserting is a major feature of an automated testing tool

Cypress is providing the major standard assertion libs out of the box :

- chai (BDD and TDD)
- chai-jquery
- sinon-chai
- common assertions (should)

Playwright is providing only 1 build-in assertion system.
Any standard other lib must be be downloaded and specifically imported.

exercise 8

goal : use an assertion

asserting

exercise 8

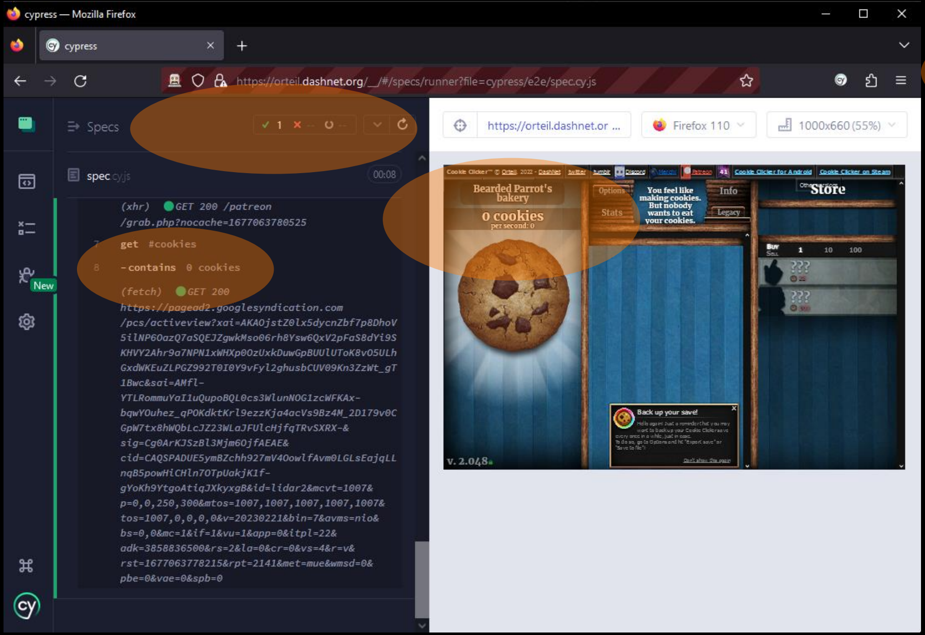
```
λ cat cypress.env.json
{
  …
  "nbcookies":"#cookies"
}

λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  …
  cy.get(Cypress.env('nbcookies'))
  .contains('0 cookie');
  …
});
```

```
λ cat .env

  …
  nbcookies="#cookies"


λ cat tests/example.spec.js
test.describe('test suite', () =>
{
  …
  await
  expect(page.locator(process.env.nbcookies))
  .toContainText('0 cookie');
  …
```

Re

asserting

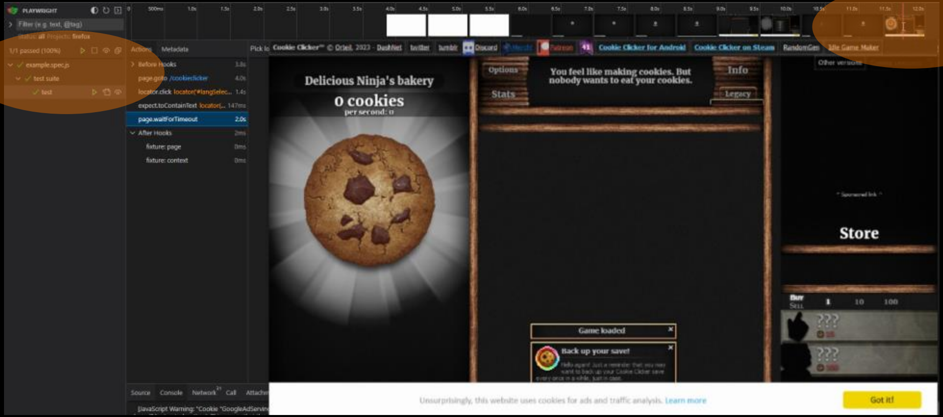
exercise 8



cypress — Mozilla Firefox

https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

```
spec.cy.js                                             00:08
(xhr) GET 200 /patreon
/grab.php?nocache=1677063780525
get  #cookies
-contains  0 cookies
(fetch) GET 200
https://pagead2.googlesyndication.com
/pcs/activeview?xai=AKAOjstZ0lx5dycnZbf7p8DhoV
5ilNP6OazQ7aSQEJZgwkMso06rh8Ysw6QxV2pFaS8dYi9S
KHVY2Ahr9a7NPN1xWHXp0OzUxkDuwGpBUUlUToK8vO5ULh
GxdWKEuZLPGZ992T0I0Y9vFyl2ghusbCUV09Kn3ZzWt_gT
1Bwc&sai=AMfl-
YTLRommuYaI1uQupoBQL0cs3WlunNOG1zcWFKAx-
bqwYOuhez_qPOKdktKrl9ezzKja4acVs9Bz4M_2D179v0C
GpW7tx8hWQbLcJZ23WLaJFUlcHjfqTRvSXRX-&
sig=Cg0ArKJSzBl3Mjm6OjfAEAE&
cid=CAQSPADUE5ymBZchh927mV4OowlfAvm0LGLsEajqLL
nqB5powHiCHln7OTpUakjK1f-
gYoKh9YtgoAtiqJXkyxgB&id=lidar2&mcvt=1007&
p=0,0,250,300&mtos=1007,1007,1007,1007,1007&
tos=1007,0,0,0,0&v=20230221&bin=7&avms=nio&
bs=0,0&mc=1&if=1&vu=1&app=0&itpl=22&
adk=3858836500&rs=2&la=0&cr=0&vs=4&r=v&
rst=1677063778215&rpt=2141&met=mue&wmsd=0&
pbe=0&vae=0&spb=0
```



```
PLAYWRIGHT
Filter (e.g. text, @tag)
Status: all  Projects: firefox
1/1 passed (100%)
✓ example.spec.js
  ✓ test suite
    ✓ test

Actions   Metadata
> Before Hooks                              3.8s
  page.goto /cookieclicker                  4.0s
  locator.click locator('#langSelec...      1.4s
  expect.toContainText locator(...          147ms
  page.waitForTimeout                       2.0s
v After Hooks                               2ms
    fixture: page                           0ms
    fixture: context                        0ms

Source  Console  Network  Call  Attachm
[JavaScript Warning: "Cookie "GoogleAdServin
```

test organization and context management

Cypress and Playwright are clearing the context by default

- the DOM
- the cookies
- the local storage

between each test !

test organization and context management

Cypress proposes 2 mechanisms for context management

- session
  - tests are isolate by default, context is saved using a cy.session call and restored using another cy.session call
  - restore is taking a bit of time and the entire test must be in an it scope
- test isolation
  - the user controls if Cypress clears the context or not for every test suite aka describe scope
  - test content can be splitted over several it calls
  - it runs faster

✓ preferred solution

test organization and context management

Playwright proposes to use the beforeAll and afterAll functions to open and close the page.

Don't forgot to run the test suite in "serial mode" to avoid Playwright running the tests in parallel.

test organization and context management

```
describe('clear env', { testIsolation:true }, () =>
{
  it('clear', () => { cy.log(‘context cleared'); });
});

describe('testing Cookie Clicker', { testIsolation:false }, () =>
{
 it('test1’, () => { // do something });
 it('test2’, () => { // do something else in the same context });
});
```

clears context

keeps context

test organization and context management

```
test.describe('test suite', () =>
{
  test.describe.configure({ mode: 'serial' });
  let page;
  test.beforeAll(async ({ browser }) => { page = await browser.newPage(); });
  test.afterAll(async () => { await page.close(); });

  test('test1', async () => { // do something with page });
  test('test2', async () => { // do something else in the same context });

  test('debug only', async () =>  { await page.waitForTimeout(5000); });
});
```

use only for debug to avoid Playwright ending the test immediately and thus preventing to see the proper screenshot at the end

exercise 9

goal : organize tests and control context

test organization and context management

exercise 9

```
λ cat cypress.env.json
{
  …
   "bigcookie":"#bigCookie"
}
```

```
λ cat .env
{
  …
   bigcookie="#bigCookie"
}
```

test organization and context management

exercise 9

```
describe('clear env', { testIsolation:true }, () => {
  it('clear', () => { cy.log('env cleared'); }); });

describe('test suite', { testIsolation:false }, () => {
  it('init', () =>  {
    cy.visit('/');
    cy.get(Cypress.env('lang')).click();
    cy.get(Cypress.env('care')).click();
    cy.get(Cypress.env('nbcookies')).contains('0 cookie'); });
  it('bake', { scrollBehavior: false }, () =>  {
    cy.get(Cypress.env('bigcookie')).click();
    cy.get(Cypress.env('nbcookies')).contains('1 cookie'); });
});
```

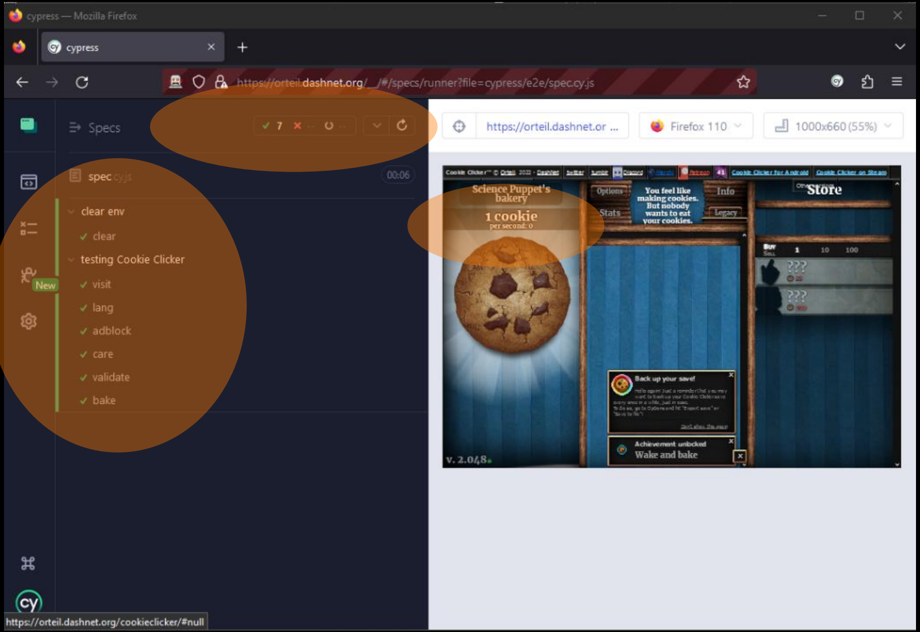
test organization and context management

exercise 9

```
test.describe('test suite', () => {
  test.describe.configure({ mode: 'serial' });
  let page;
  test.beforeAll(async ({ browser }) => { page = await browser.newPage(); });
  test.afterAll(async () => { await page.close(); });
  test('init', async () => {
    await page.goto('/cookieclicker');
    await page.locator(process.env.lang2).click();
    await page.locator(process.env.care).click();
    await expect(page.locator(process.env.nbcookies)).toContainText('0 cookie'); });
  test('bake', async () => {
    await page.locator(process.env.bigcookie).click();
    await expect(page.locator(process.env.nbcookies)).toContainText('1 cookie'); });
});
```

test organization and context management

exercise 9



cypress — Mozilla Firefox

cypress

https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

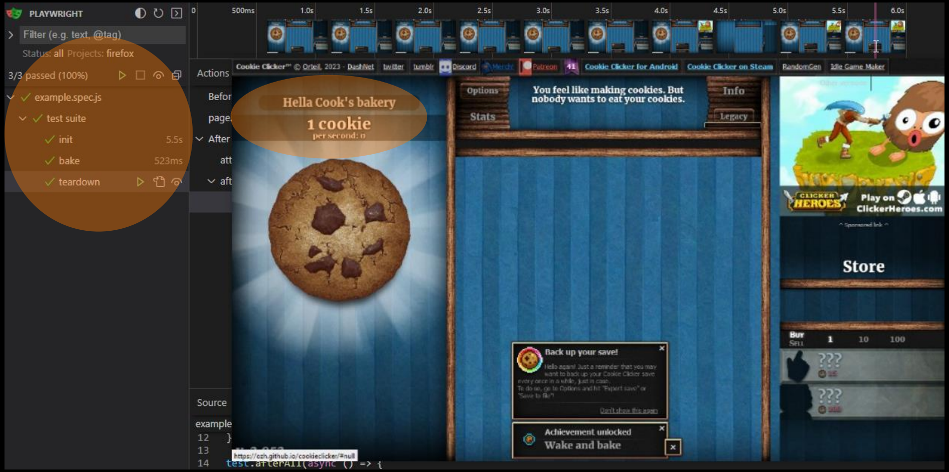
Specs ✓ 7 ✕ --

spec.cy.js 00:06

- clear env
  - ✓ clear
- testing Cookie Clicker
  - ✓ visit
  - ✓ lang
  - ✓ adblock
  - ✓ care
  - ✓ validate
  - ✓ bake

https://orteil.dashnet.or ... Firefox 110 1000x660 (55%)

https://orteil.dashnet.org/cookieclicker/#null



PLAYWRIGHT

Filter (e.g. text, @tag)

Status: all Projects: firefox

3/3 passed (100%)

- ✓ example.spec.js
  - ✓ test suite
    - ✓ init 5.5s
    - ✓ bake 523ms
    - ✓ teardown

Actions

Source

```
}
test.afterAll(async () => {
```

https://orteil.github.io/cookieclicker/#null

keyboard interaction

whether it is for filling a form or any other keyboard usage

```
cy.get(selector).type('{ctrl}s');
```

```
await
page.keyboard.press('Control+s');
```

press and type are different in Playwright !

exercise 10

goal : simulate keyboard action

keyboard interaction

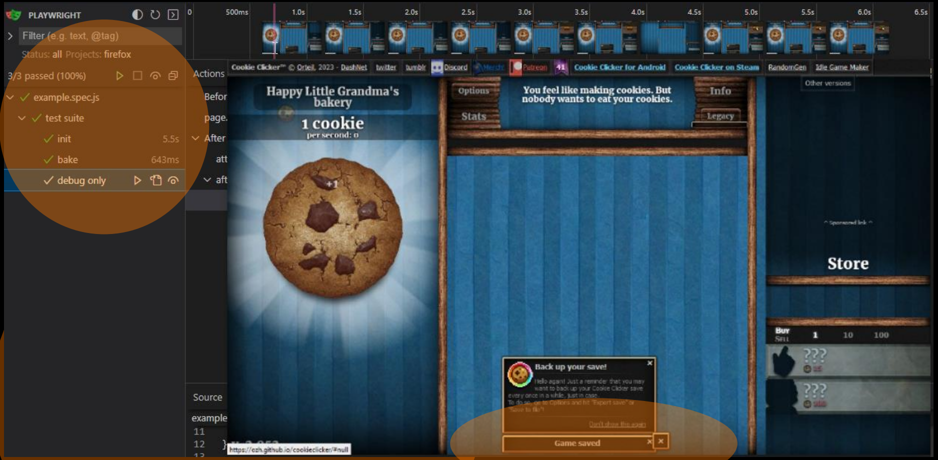
exercise 10

```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  …
  it('save', () =>
  {
    cy.get('body').type('{ctrl}s');
  });
});
```

```
λ cat tests/example.spec.js
test.describe('test suite', () =>
{
  …
  test('save', async () =>
  {
    await
page.keyboard.press('Control+s');
  });
});
```

Cypress : keyboard interaction

exercise 10


PLAYWRIGHT
Filter (e.g. text, @tag)
Status: all Projects: firefox
3/3 passed (100%)
example.spec.js
test suite
init 5.5s
bake 643ms
debug only

Actions
Befor
page.
After
att
aft
Source
example
Happy Little Grandma's bakery
1 cookie
per second: 0
Options
Stats
You feel like making cookies. But nobody wants to eat your cookies.
Info
Legacy
Store
Back up your save!
Game saved
https://ozh.github.io/cookieclicker/#null

Cypress : reload

reload the current page

```
cy.reload();

cy.reload(true);
// clears the cache
```

```
await page.reload();
```

exercise 11

goal : reload the current page

reload

exercise 11

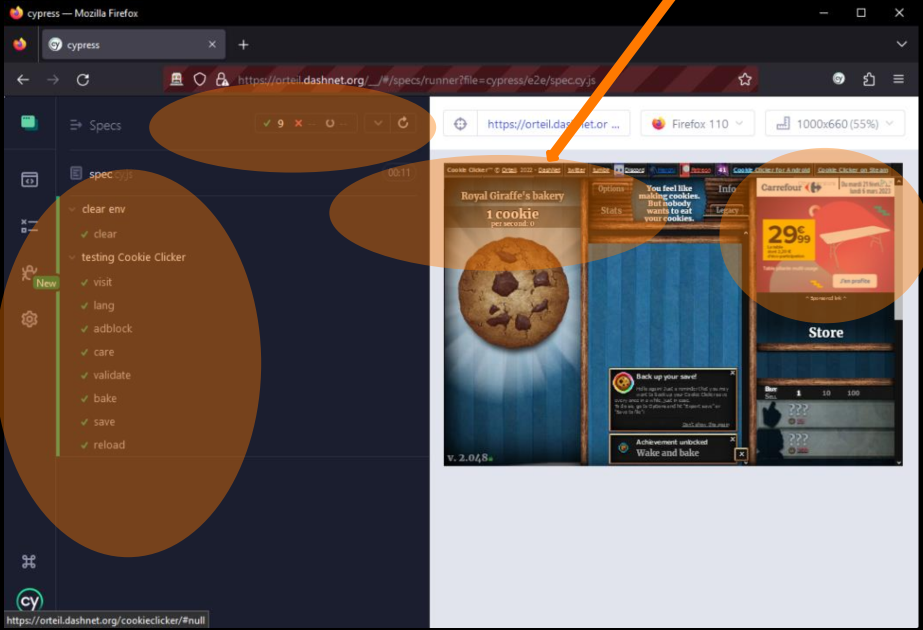
```
λ cat cypress/e2e/spec.cy.js
describe('test suite', () =>
{
  …
  it('reload', () =>
  {
    cy.reload();
  });
  …
});
```

```
λ cat tests/example.spec.js
test.describe('test suite', () =>
{
  …
  test('reload', async () =>
  {
    await page.reload();
  });
  …
});
```

reload

note that the éléments linked to the context (cookie, local storage) are recovered

exercise 11



cypress — Mozilla Firefox

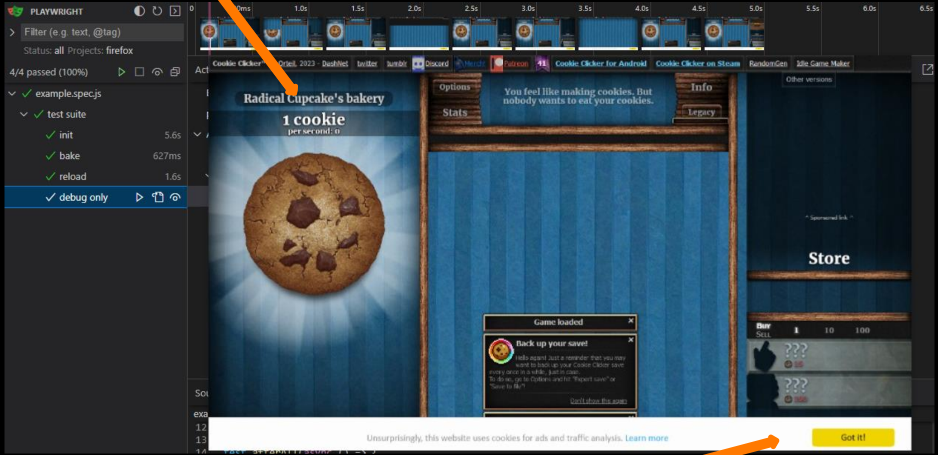
https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

Specs ✓ 9 ✕ --

spec.cy.js 00:11

- clear env
  - ✓ clear
- testing Cookie Clicker
  - ✓ visit
  - ✓ lang
  - ✓ adblock
  - ✓ care
  - ✓ validate
  - ✓ bake
  - ✓ save
  - ✓ reload

https://orteil.dashnet.org/cookieclicker/#null



PLAYWRIGHT

Filter (e.g. text, @tag)

Status: all Projects: firefox

4/4 passed (100%)

- ✓ example.spec.js
  - ✓ test suite
    - ✓ init 5.6s
    - ✓ bake 627ms
    - ✓ reload 1.6s
    - ✓ debug only

problem in Playwright, cookies are not restored ?

scrolling

to simulate scrolling :

```
cy.scrollTo(position|x,y);

cy.scrollIntoView();
```

```
await page.evaluate(() =>
   document
   .querySelector(locator)
   .scrollBy(0, 1000)
);
```

nothing specific in Playwright,
fallback on standard js, a bit awkward …

exercise 12

goal : scroll the page

scrolling

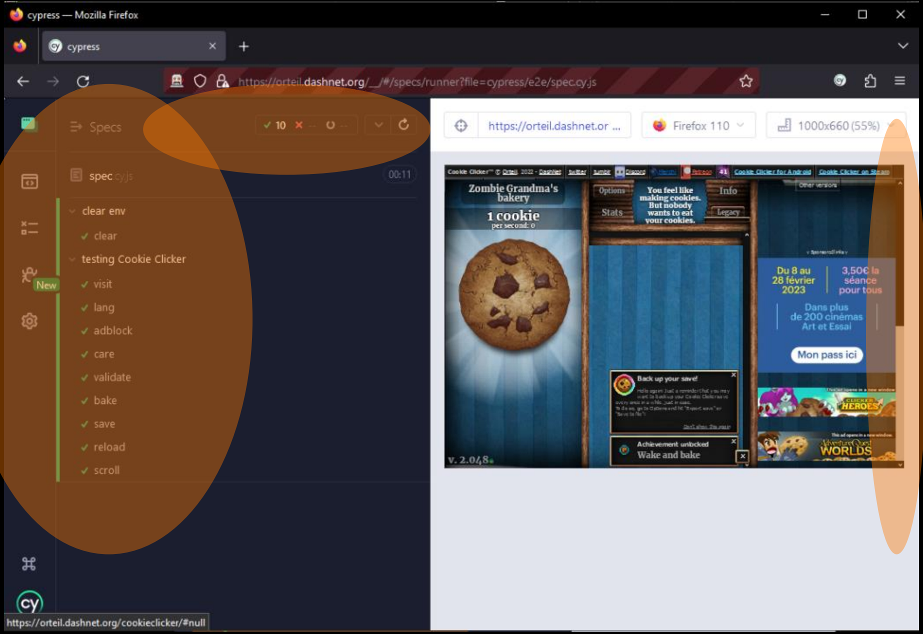
exercise 12

```
λ cat cypress.env.json
{
  …
  "sectionright":"#sectionRight"
}
λ cat cypress/e2e/spec.cy.js
describe('test suite', () => {
  …
  it('scroll', () => {
    cy.get(Cypress.env('sectionright'))
    .scrollTo('bottom');
  });
});
```

```
λ cat .env
{
  …
  sectionright="#sectionRight"
}
λ cat tests/example.spec.js
describe('test suite', () => {
  …
  await page.evaluate(() =>
    document.querySelector('#sectionRight')
              .scrollBy(0, 1000))
  });
});
```

Cypress : scrolling

exercise 12



cypress — Mozilla Firefox

https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js

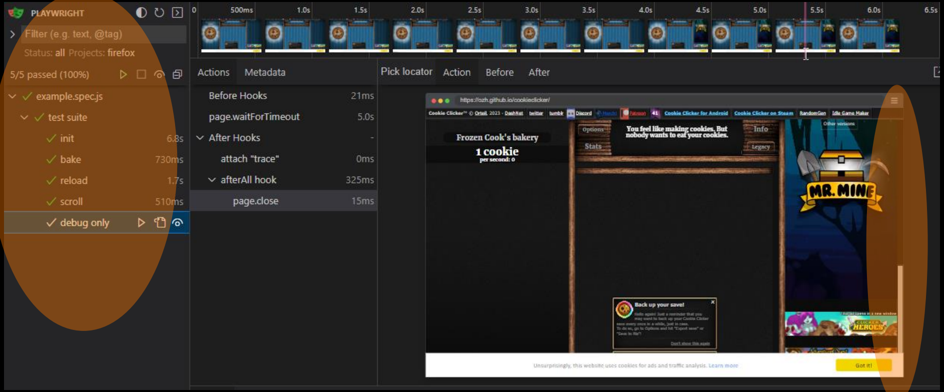
Specs ✓ 10 ✕ -- ○ --

spec.cy.js 00:11

- clear env
  - ✓ clear
- testing Cookie Clicker
  - ✓ visit
  - ✓ lang
  - ✓ adblock
  - ✓ care
  - ✓ validate
  - ✓ bake
  - ✓ save
  - ✓ reload
  - ✓ scroll

https://orteil.dashnet.or... Firefox 110 1000x660 (55%)

https://orteil.dashnet.org/cookieclicker/#null



PLAYWRIGHT

Filter (e.g. text, @tag)

Status: all Projects: firefox

5/5 passed (100%)

- ✓ example.spec.js
  - ✓ test suite
    - ✓ init 6.8s
    - ✓ bake 730ms
    - ✓ reload 1.7s
    - ✓ scroll 510ms
    - ✓ debug only

Actions Metadata

Action	Duration
Before Hooks	21ms
page.waitForTimeout	5.0s
After Hooks	-
attach "trace"	0ms
afterAll hook	325ms
page.close	15ms

Pick locator Action Before After

https://ozh.github.io/cookieclicker/

mobile web

to simulate a mobile web device :

set the viewport size

```
cy.viewport(screen);
```

set the user agent

```
cy.intercept('/', (req) =>
req.headers['user-agent'] = useragent);
```

set the viewport size

```
 test.use({ viewport:
 { width: 123, height: 456 }});
```

set the user agent

```
test.use({ userAgent: useragent });
```

exercise 13

goal : simulate a mobile browser

mobile web

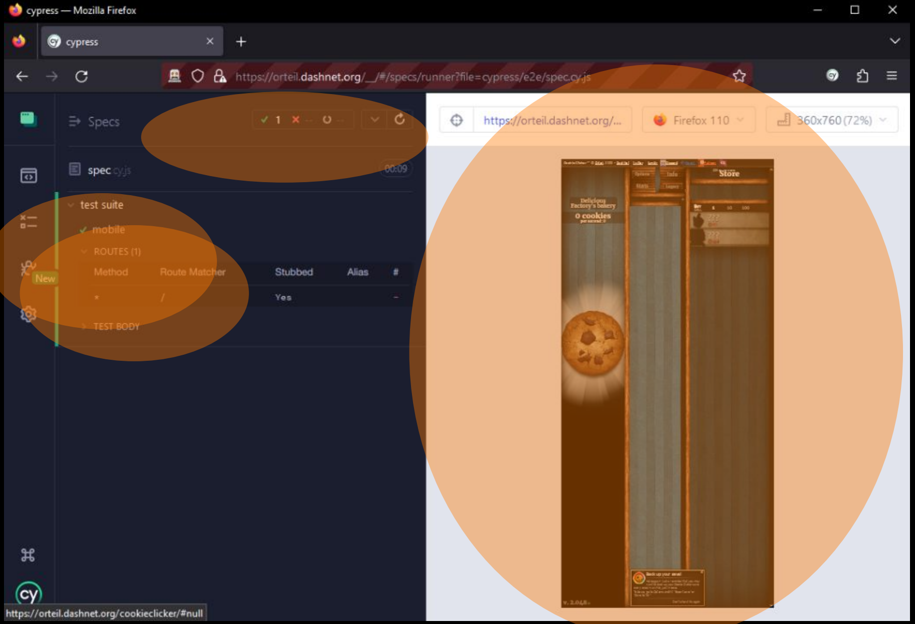
exercise 13

```
λ cat cypress/e2e/spec.cy.js
const config = {screen:'samsung-s10',
                useragent:"Mozilla/5.0 (Android 9;
Mobile; rv:97.0) Gecko/20100101 Firefox/97.0"};


describe('test suite', () => {
  it('mobile', () => {
    cy.viewport(config.screen);
    cy.intercept('/', (req) => req.headers['user-
agent'] = config.useragent);
    cy.visit('/');
    cy.get(Cypress.env('lang')).click();
    cy.get(Cypress.env('care')).click();
    cy.get(Cypress.env('nbcookies')).contains('0
cookie');
  }); });
```

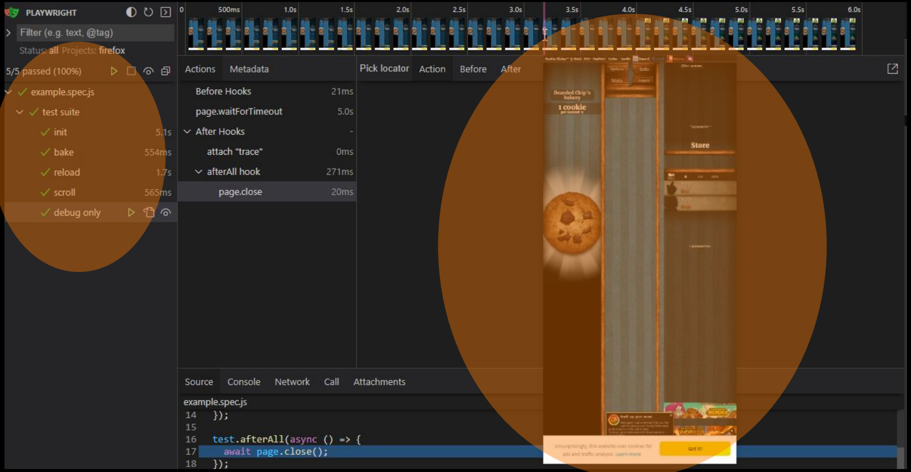
```
λ cat tests/example.spec.js
const config = {screen:{width:360, height:760},
                useragent:"Mozilla/5.0 (Android 9;
Mobile; rv:97.0) Gecko/20100101 Firefox/97.0"};


test.describe('test suite', () => {
  test.use({ userAgent: config.useragent });
  test.use({ viewport: config.screen });
  test('test', async ({page}) =>  {
    await page.goto('/cookieclicker');
    await page.locator(process.env.lang2).click();
    await page.locator(process.env.care).click();
    await expect(page.locator(process.env.nbcookies))
          .toContainText('0 cookie');
    await page.waitForTimeout(20000);
  });});
```

mobile web

exercise 13



cypress — Mozilla Firefox
cypress
https://orteil.dashnet.org/__/#/specs/runner?file=cypress/e2e/spec.cy.js
Specs
spec.cy.js
00:09
test suite
mobile
ROUTES (1)
Method
Route Matcher
Stubbed
Alias
#
Yes
TEST BODY
https://orteil.dashnet.org/...
Firefox 110
360x760 (72%)
https://orteil.dashnet.org/cookieclicker/#null



```
  });

  test.afterAll(async () => {
    await page.close();
  });
```

wrap-up

to wrap-up, I propose a exercice to simulate really playing Cookie Clicker ☺

exercise 14

goal : play Cookie Clicker

play Cookie Clicker

exercise 14

1- start with code for the exercice 9

2- click on the big cookie until you have enough cookies to buy a cursor, then buy the cursor

3- click on the big cookie until you have enough cookies to buy a grandma or an upgrade, then buy the latest

4- carry on until you have plenty of cookies ☺

Thanks