

CYPRESS

Mats Fischer

WHAT IS CYPRESS?

Introduction, goals, key features, advantages

CYPRESS

What is Cypress?

- (End-to-end) Testing framework
- Built for modern web applications
- Open-source

“Cypress can test anything that runs in a browser”

~ Cypress Docs

KEY FEATURES & GOALS

Key features

- Setting up, writing, running and debugging tests
- Allows End-to-end, component, integration and unit tests

Goals

- Easy to use (Application & Documentation)
- Flat learning curve
- Good community

“enhance productivity, make testing enjoyable and generate developer happiness” ~ Cypress Docs

WHY CYPRESS?

Easy usage

- Easy setup and installation
- Interactive & intuitive test runner
- Fast and reliable testing
- Fast test execution
- Simplified API for writing tests
- Real-time reloading for rapid development

WHY CYPRESS?

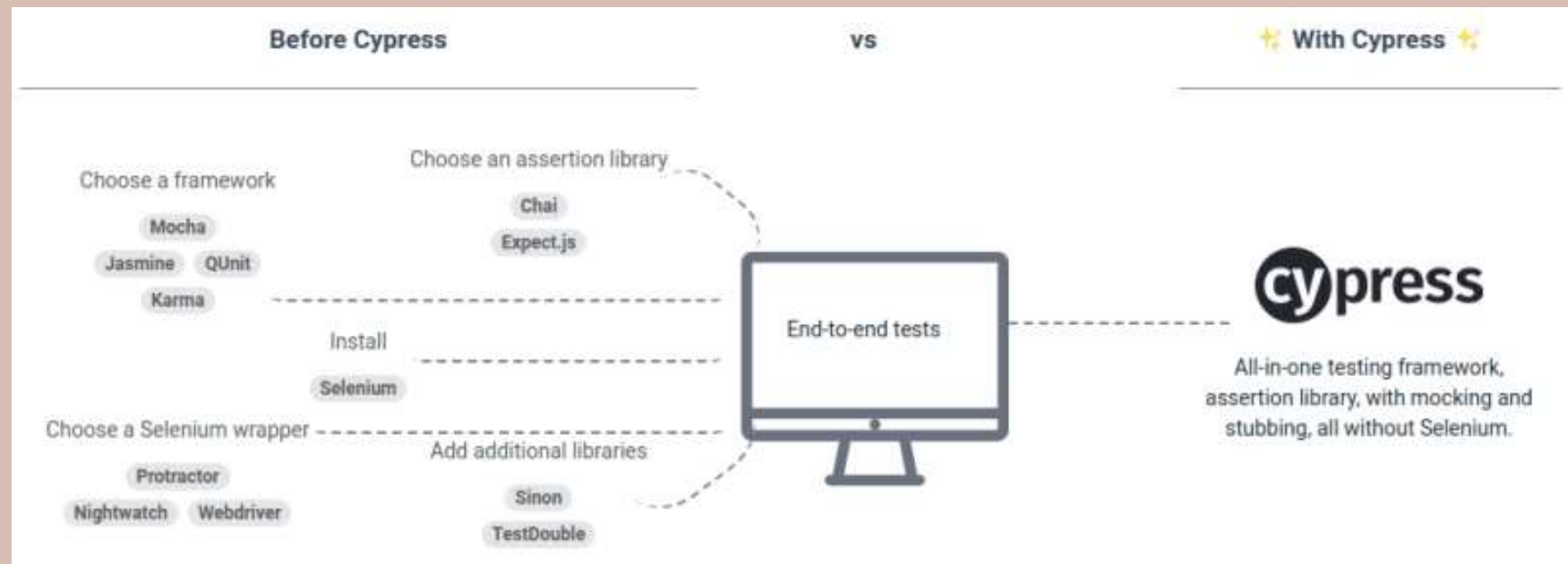
Usefull features

- Automatic waiting
- Easy debugging with “time-travel“, known debugging tools and readable errors
- Network traffic control
- ...

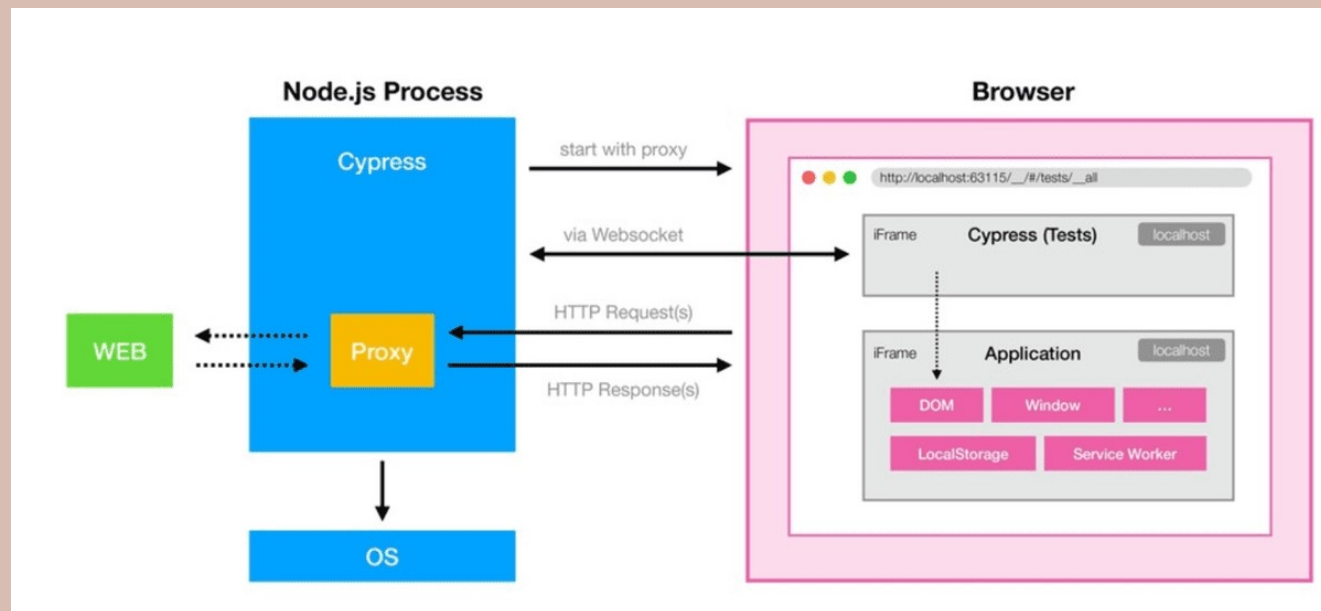
Ecosystem

- Support for modern web technologies (React, Angular, Vue, ...)
- Documentation and community support

CYPRESS ARCHITECTURE



CYPRESS ARCHITECTURE



RESSOURCEN & COMMUNITY

Ressourcen

- Amazing [https://docs.cypress.io/
Documentation: api/table-of-contents/](https://docs.cypress.io/api/table-of-contents/)

Community

- Cypress Plugins
- Forums

INTEGRATION & PLUGINS

Intergration

- Continuous Integration
- Reporting & code-coverage tools

Plugins

- Cypress is open-source!

CODE STRUCTURE

How to write code for Cypress

INSTALLING CYPRESS

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

```
npx cypress open
```

TEST FILE STRUCTURE

ARRANGE → ACT → ASSERT

```
describe("React TodoMVC", () => {  
  it("adds a single todo", () => {  
    cy.visit("http://localhost:8888") // Arrange  
    cy.get(".new-todo").type("Buy Milk{enter}") // Act  
    cy.get(".todo-list li").should("have.length", 1) // Assert  
  })  
})
```



CSS SELECTORS

COMMAND CHAINING

```
cy.get(".todo-list li").find("label").should("contain", "Buy Milk")
```

ANY KEYWORDS FROM CHAI



SETUP & CLEAN UP

```
describe( title: 'Home Page', fn: () => {  
  beforeEach( fn: () => {  
    // ran before each test to setup  
  })  
  
  it( title: 'tests something', config: () => {  
    // testing...  
  })  
  
  afterEach( fn: () => {  
    // ran after each test to clean up  
  })  
})
```

CUSTOM COMMANDS

```
cy.getByData("hero-heading")
```

instead of:

```
cy.get("[data-test='hero-heading']")
```


CUSTOM COMMANDS

COMMANDS.JS:

```
Cypress.Commands.add("getByData", (selector) => {  
  return cy.get(`[data-test=${selector}]`)  
})
```

FIXTURES

ANY JSON FILES FROM CYPRESS DIRECTORY

```
cy.fixture("users.json").as("userData")  
  
cy.get("username").type(userData[0].username)  
cy.get("password").type(userData[0].password)
```

SENDING REQUESTS

```
cy.request("POST", "http://localhost:8888/users/admin", { name: "Jane" }).then(
  (response) => {
    // response.body is automatically serialized into JSON
    expect(response.body).to.have.property("name", "Jane") // true
  }
)
```

INTERCEPTING REQUESTS

```
cy.intercept("GET", "/transactions/public*", {  
  headers: {  
    "X-Powered-By": "Express",  
    Date: new Date().toString(),  
  },  
})
```

IMPORTANT COMMANDS

- `cy.get(selector)` // get DOM element
- `cy.title()` // get Title of HTML Document
- `.type(string)` // type into a HTML-Element
- `.click()` // click a HTML-Element
- `.find(selector)` // similar to get. Searches HTML-Element for another
- `.eq(index)` // select one item from multiple
- `.should(chainer, value)` // assert something

For further information refer to the Cypress documentation (<https://docs.cypress.io/api/table-of-contents/>)

```
npm install
```

```
npx cypress open
```

HANDS ON

Now try yourself!

Download the latest commit of the repository and follow the instructions in [“cypress/e2e/spec.cy.js”](#)

THANK
YOU!

