

Less painful E2E tests with Cypress.io

Pavel Kruhlei, FOSSDEM 2019

About me



Quality Engineering Lead in fintech startup in Amsterdam



10+years in QA. 5 in automation.

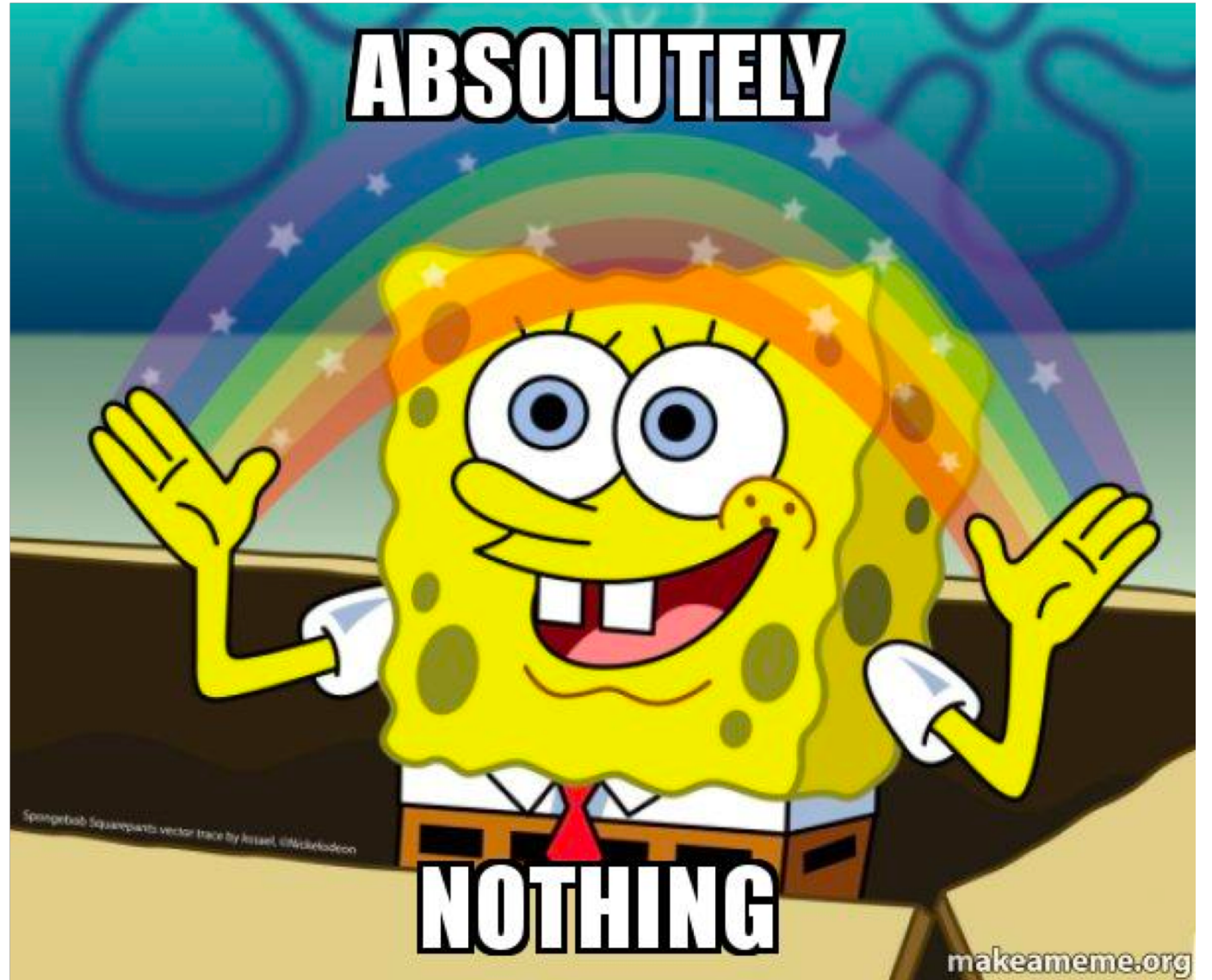


DevTestSecOpsWhatewer, big fan of serverless

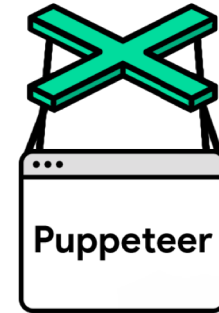


Free time: bouldering, drinking Belgium beer

What's wrong
with
Selenium?



What else?



What is Cypress?



JS (TS) framework for test automation

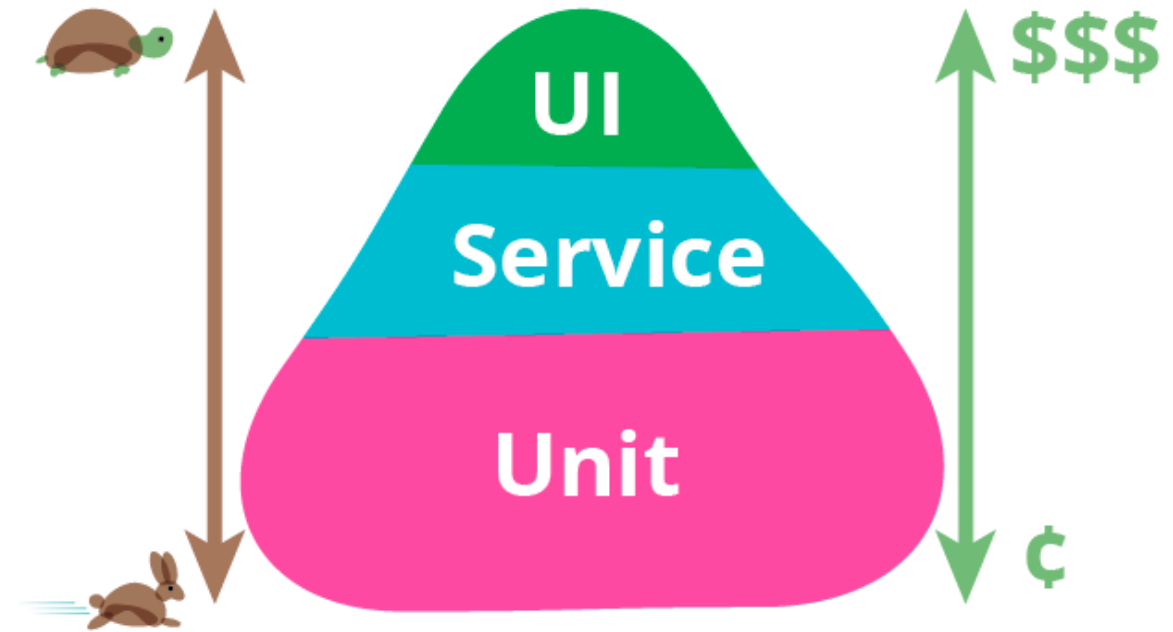
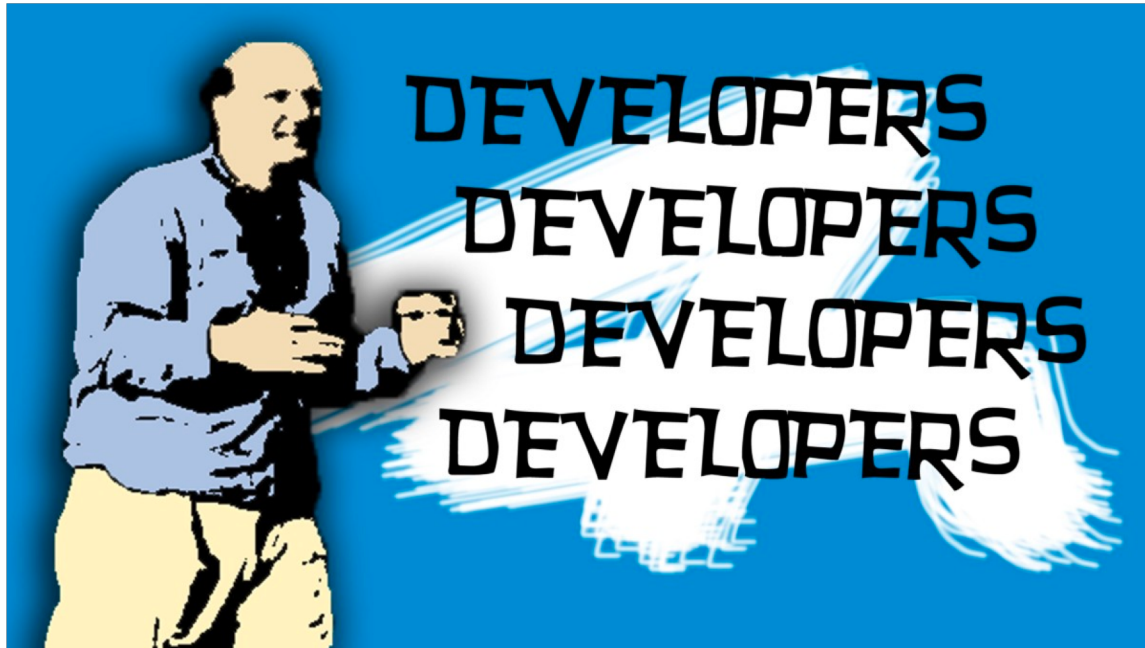


Runner/Debugger app



SAAS for reporting and balancing

Why Cypress?



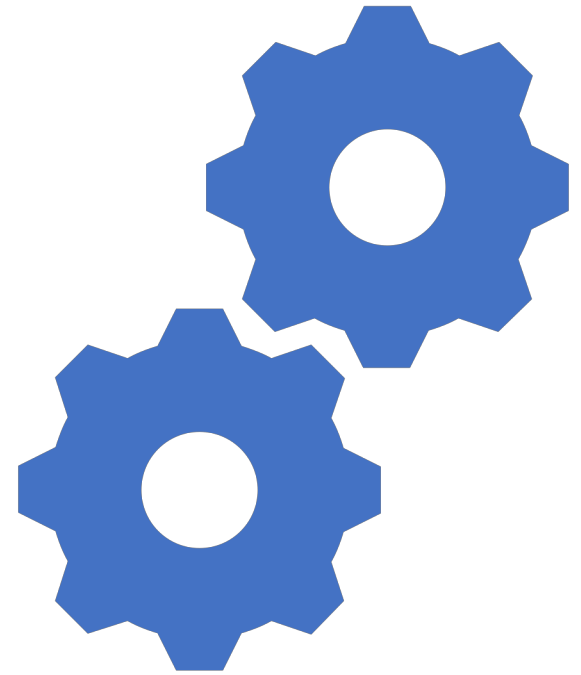
Cypress and it's dev-obsession

- Bundled tools
- Reporters (junit, teamcity etc)



Cypress and it's dev-obsession

- Extendable by plugins
- Easy setup
- CI-ready
- Debugging
- Stubs, spies and clocks
- Automatic waits



Test
layers

Unit

Integrstion

E2E



Unit tests

```
1 describe('File upload', () => {
2   beforeEach(() => {
3     cy.visit('index.html')
4   })
5
6   it('uploads mock file and stubs the server', () => {
7     // see how to mock a remote server
8     // https://on.cypress.io/route
9     cy.server()
10    cy.route('POST', 'https://some-server.com/upload', 200).as('upload')
11    // load mock data from a fixture or construct here
12    const testFile = new File(['data to upload'], 'upload.txt')
13    cy.get('#file1').trigger('change', { testFile })
14    // make sure upload has happened
15    cy.wait('@upload')
16  })
17 }
```

Integration test

```
describe('Logging In - HTML Web Form', function(){
  context('Unauthorized', function(){
    it('is redirected on visit to /dashboard when no session', function(){
      // we must have a valid session cookie to be logged
      // in else we are redirected to /unauthorized
      cy.visit('/dashboard')
      cy.get('h3').should('contain', 'You are not logged in and cannot access this page')
      cy.url().should('include', 'unauthorized')
    })

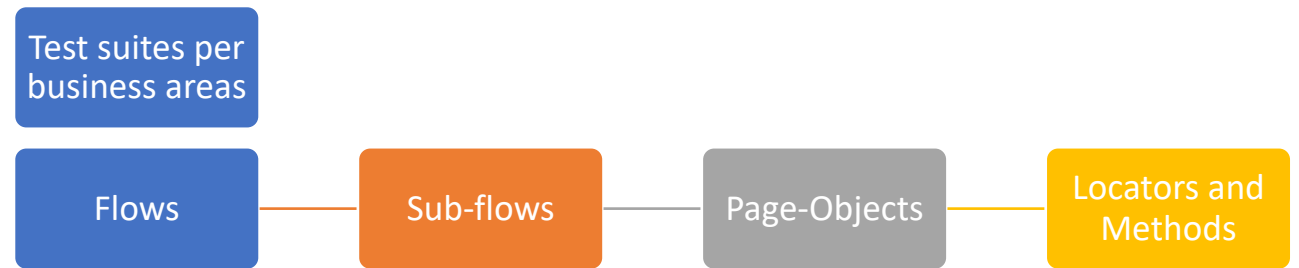
    it('is redirected using cy.request', function(){
      // instead of visiting the page above we can test this by issuing
      // a cy.request, checking the status code and redirectedToUrl property.

      // See docs for cy.request: https://on.cypress.io/api/request

      // the 'redirectedToUrl' property is a special Cypress property under the hood
      // that normalizes the url the browser would normally follow during a redirect
      cy.request({
        url: '/dashboard',
        followRedirect: false // turn off following redirects automatically
      })
      .then((resp) => {
        // should have status code 302
        expect(resp.status).to.eq(302)

        // when we turn off following redirects Cypress will also send us
        // a 'redirectedToUrl' property with the fully qualified URL that we
        // were redirected to.
        expect(resp.redirectedToUrl).to.eq('http://localhost:7077/unauthorized')
      })
    })
  })
})
```

E2E Tests (how we build it)



Traid- offs

Chrome-only

Its a test automation tool still, not a
Swiss army knife (no spider,
performance tests or 3rd party tests)

Single-browser, single tab

Same domain pages

Cypress test strategy

Stub/not Stub

Deterministic tests

Independent tests

No XPATH

Control application state

Do not test 3rd parties

Multiple assertions are fine

WRITING TESTS IS LIKE SEX



MORE YOU DO - BETER YOU GET AT IT



It's QUESTION TIME!!