Cypress at scale

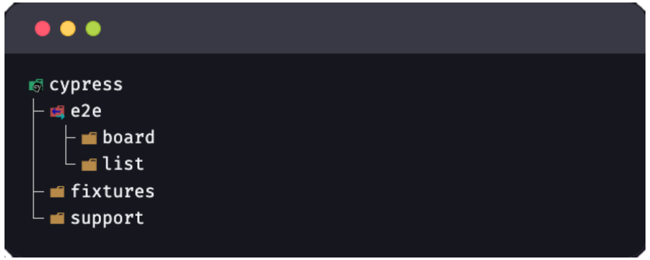
Filip Hric

- Filip Hric is a DevRel at Replay.io
- He teaches testers about web development and developers about testing.
- Filip is a Cypress.io ambassador, leads a "Learn Cypress.io" community on Discord and he has a blog at filiphric.com where he publishes Cypress.io tips.
- He has taught hundreds of testers and developers about good practices and advanced concepts for testing. He enjoys running, playing guitar and spending time with his wife and four children.

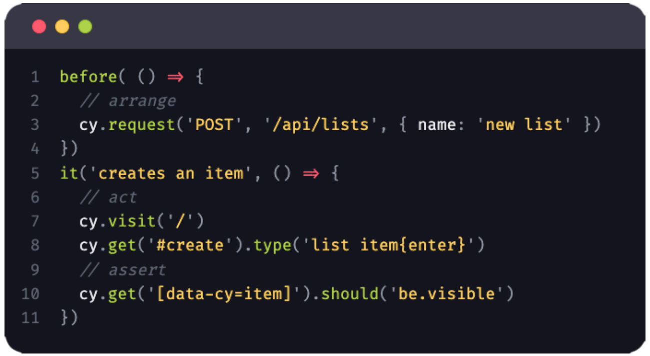
Test design

- focusing on behaviors
- arrange - act - assert
- testing patterns

  - POM?
  - BDD?
  - DRY?
  - Custom commands?



```
cypress
├ e2e
│ ├ board
│ └ list
├ fixtures
└ support
```



```
before( () => {
  // arrange
  cy.request('POST', '/api/lists', { name: 'new list' })
})
it('creates an item', () => {
  // act
  cy.visit('/')
  cy.get('#create').type('list item{enter}')
  // assert
  cy.get('[data-cy=item]').should('be.visible')
})
```

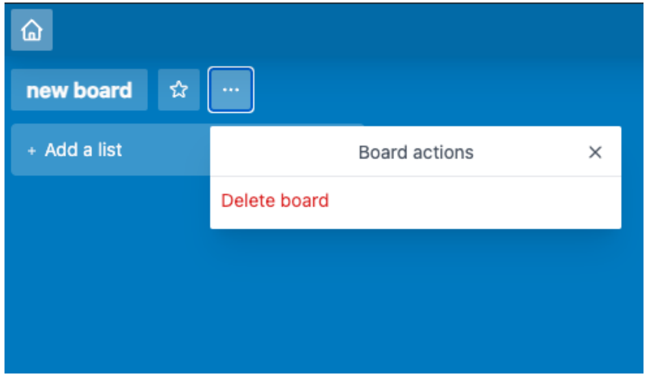
Test design

test:

```
it('opens menu', () => {
  cy.visit('/')
  cy.openMenu()
})
```

custom command:

```
Cypress.Command.add('openMenu', () => {
  cy.get('[data-test="menu"]')
    .click()
    .find('[data-test="menu-item"]')
    .click()
})
```



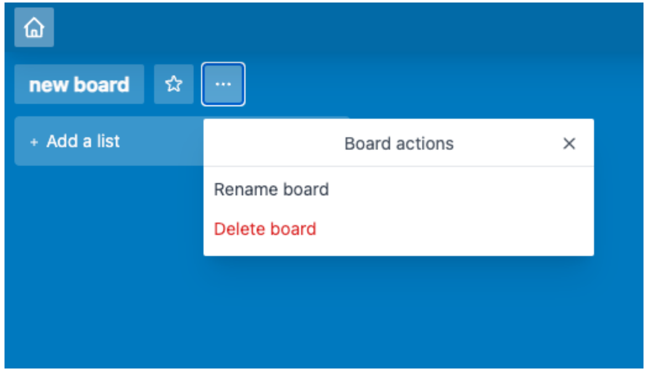
Test design

test:

```
it('opens menu', () ⇒ {
  cy.visit('/')
  cy.openMenu("Delete board")
})
```

custom command:

```
Cypress.Command.add('openMenu', (item) ⇒ {
  cy.get('[data-test="menu"]')
    .click()
    .find('[data-test="menu-item"]')
    .contains(item)
    .click()
})
```



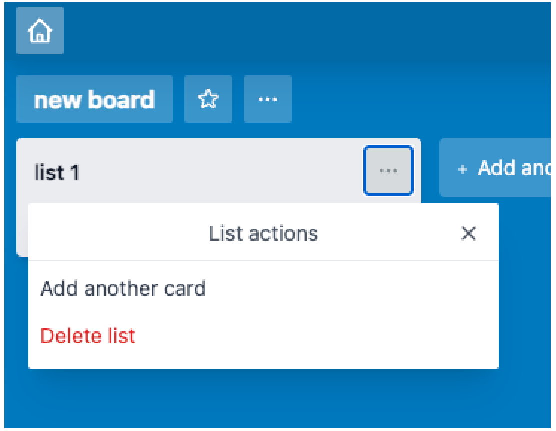
Test design

test:

```
it('opens menu', () => {
  cy.visit('/')
  cy.openMenu("Delete board", true)
})
```

custom command:

```
Cypress.Command.add('openMenu', (item, isList = false) => {
  const area = isList ? '[data-test="list-area"]' : '[data-test="header"]'
  cy.get(`${area} [data-test="menu"]`)
    .click()
    .find('[data-test="menu-item"]')
    .contains(item)
    .click()
})
```



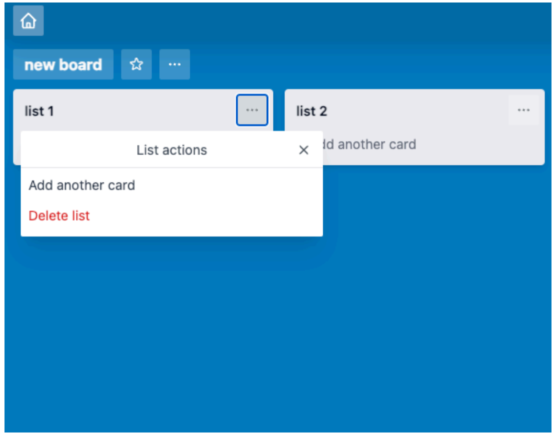
Test design

test:

```
it('opens menu', () => {
  cy.visit('/')
  cy.openMenu("Delete board", true, 0)
})
```

custom command:

```
Cypress.Command.add('openMenu', (item, isList = false, index = 0) => {
  const area = isList ? '[data-test="list-area"]' : '[data-test="header"]'
  cy.get(`${area} [data-test="menu"]`)
    .eq(index)
    .click()
    .find('[data-test="menu-item"]')
    .contains(item)
    .click()
})
```

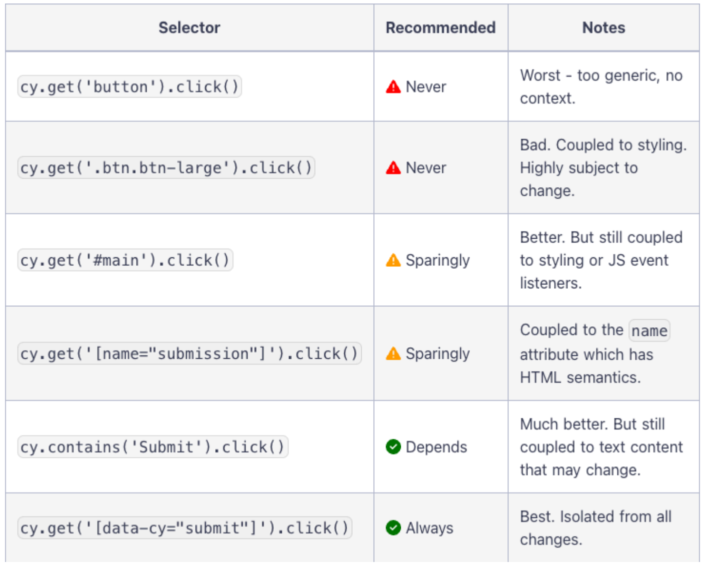


Readability

- selectors
- test annotation

```
// ❌ don't do it like this
it('board is visible', () ⇒ {})
it('works in edge cases', () ⇒ {})
it('handles input', () ⇒ {})
```

- documentation
  - installation of the projects
  - explanations, recommendations, examples
  - PR rules



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.

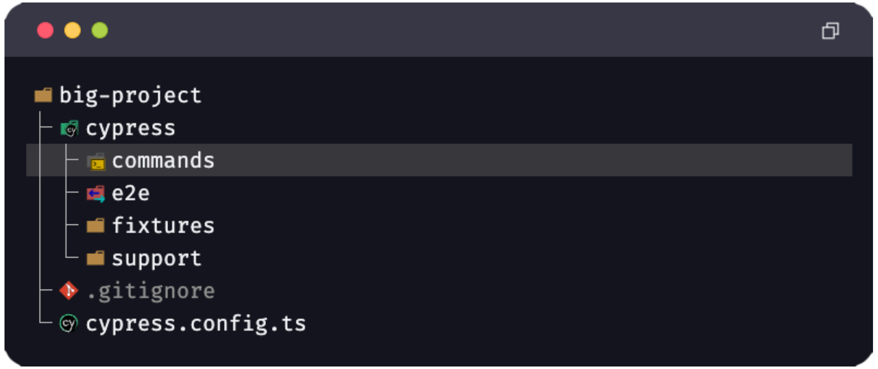
Project design

- custom commands
  - utility commands
  - API calls
  - action sequences
- typescript
- organizing custom commands
- code coverage

```
declare global {
  namespace Cypress {
    interface Chainable {
      addBoardApi: typeof addBoardApi;
    }
  }
}

/**
 * Creates a new board using the API
 * @param name name of the board
 * @example
 * cy.addBoardApi('new board')
 *
 */
export const addBoardApi = function(
  this: any, name: string
): Cypress.Chainable<any> {

  return cy
    .request('POST', '/api/boards', { name })
    .its('body', { log: false }).as('board');
```

Project design

- custom commands
  - utility commands
  - API calls
  - action sequences
- typescript
- organizing custom commands
- code coverage



```
big-project
├ cypress
│ ├ commands
│ ├ e2e
│ ├ fixtures
│ └ support
├ .gitignore
└ cypress.config.ts
```

Running in CI

- tagging tests
- configuration
- running in parallel

tagging tests:

```
it('creates a new board', { tags: ['@smoke'] }, () => {
  // test
})
```

running tags:

```
npx cypress run --env grepTags='@smoke'
```

Running in CI

- tagging tests
- configuration
- running in parallel

configuration:

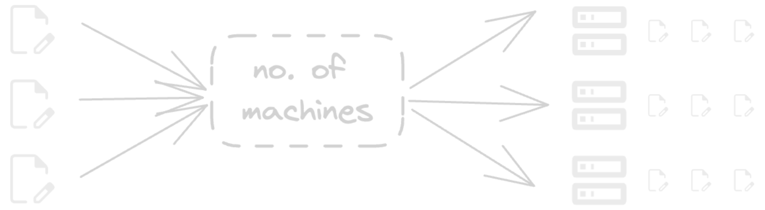
```
import { defineConfig } from 'cypress'

export default defineConfig({
  // other config attributes
  setupNodeEvents(on, config) {
    // if version not defined, use local
    const version = config.env.version || 'local'
    // load env from json
    config.env = require(`./cypress/config/${version}.json`);
    // change baseUrl
    config.baseUrl = config.env.baseUrl

    return config
  }
})
```

running tests:

Running in CI

- tagging tests
- configuration
- running in parallel

cypress-split



running:

```
npx cypress run --env split=true
```

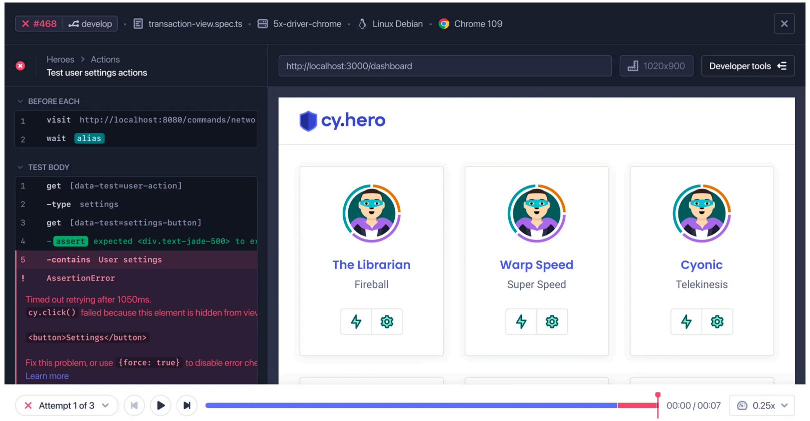
Maintenance

```
cy.contains('01')
cy.contains('04')
```

- maintenance vs. development cost
- flaky tests

Maintenance

- maintenance vs. development cost
- flaky tests



BEFORE EACH

```
visit  http://localhost:8080/commands/netwo
wait  alias
```

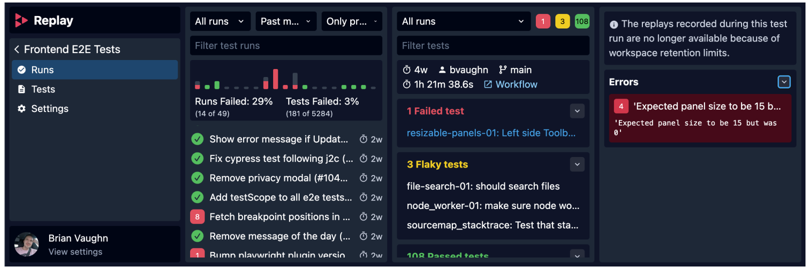
TEST BODY

```
get  [data-test=user-action]
-type  settings
get  [data-test=settings-button]
-assert  expected <div.text-jade-500> to ex
-contains  User settings
!  AssertionError
```

Timed out retrying after 1050ms.
`cy.click()` failed because this element is hidden from view

`<button>Settings</button>`

Fix this problem, or use `{force: true}` to disable error che
Learn more



Runs Failed: 29%
(14 of 49)

Tests Failed: 3%
(181 of 5284)

- ✓ Show error message if Updat... 2w
- ✓ Fix cypress test following j2c (... 2w
- ✓ Remove privacy modal (#104... 2w
- ✓ Add testScope to all e2e tests... 2w
- 8 Fetch breakpoint positions in ... 2w
- ✓ Remove message of the day (... 2w
- 1 Bump playwright plugin versio 2w

4w bvaughn main
1h 21m 38.6s Workflow

1 Failed test
resizable-panels-01: Left side Toolb...

3 Flaky tests
file-search-01: should search files
node_worker-01: make sure node wo...
sourcemap_stacktrace: Test that sta...

108 Passed tests

The replays recorded during this test run are no longer available because of workspace retention limits.

Errors

4 'Expected panel size to be 15 b...

```
'Expected panel size to be 15 but was
0'
```

What is flakiness?

Harm caused by flakiness

- loss of confidence
- more manual work
- delivery slow down
- lost test coverage
- execution slow-down
- added debugging time
- lost trust in QA process

Flaky test "fixing" strategies

muting a test

quarantining a test

retrying a test

adding a longer timeout

reproducing a test locally

deleting a test

Flaky test "fixing" strategies

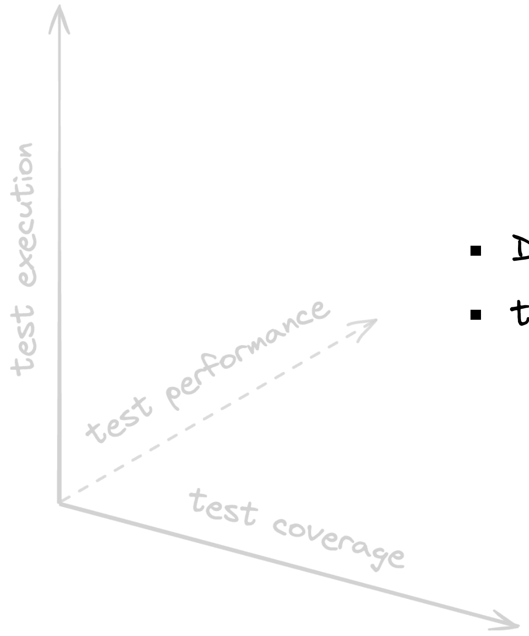
muting a test	$\longrightarrow$	shock
quarantining a test	$\longrightarrow$	denial
retrying a test	$\longrightarrow$	anger
adding a longer timeout	$\longrightarrow$	bargaining
reproducing a test locally	$\longrightarrow$	depression
deleting a test	$\longrightarrow$	acceptance

Fixing the damn problem!

or you know... not creating the problem in the first place

Test automation scaling

- isolation issues
- infrastructure issues



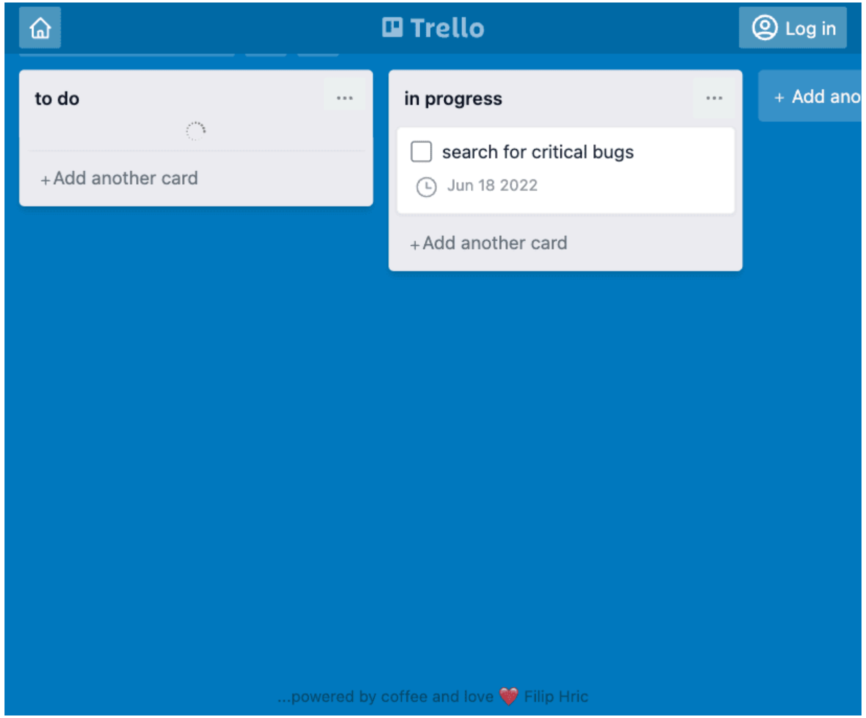
- DRY principle
- test idling

- DOM flakiness
- network flakiness

test coverage

DOM flakiness

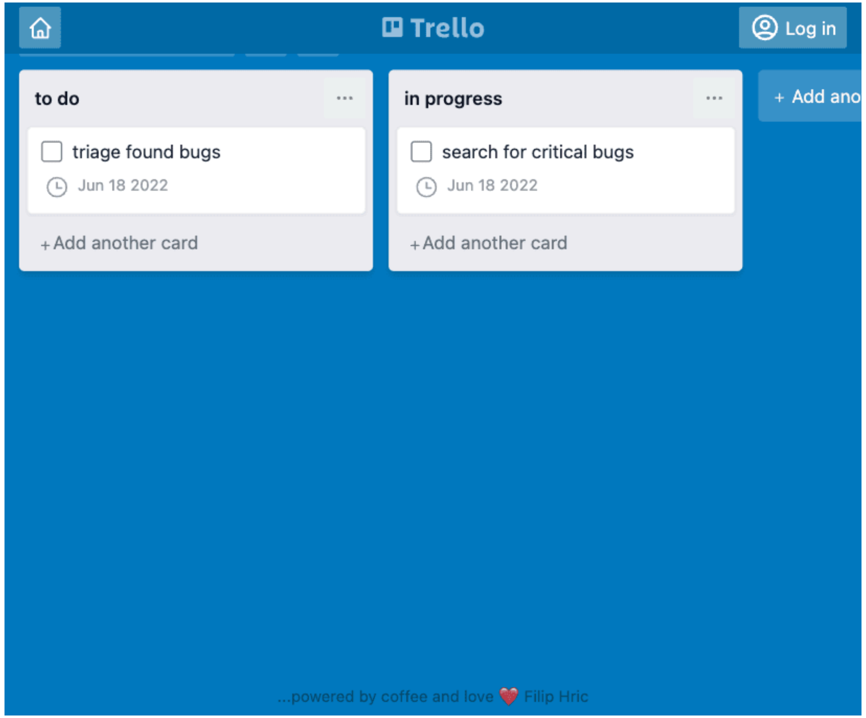
```
it('finds the word "bugs" on a card item', () ⇒ {
  cy,visit('/board/1')
  cy.get('[data-cy=card]')
    .eq(0)
    .should('contain.text', 'bugs')
})
```



DOM flakiness

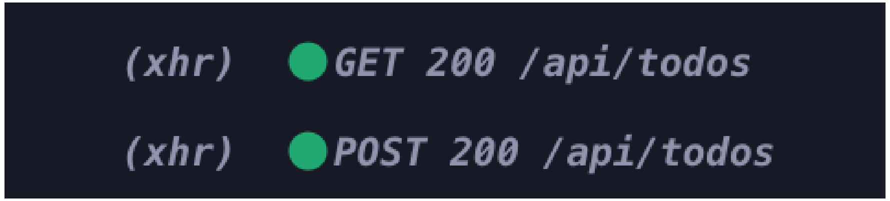
DOM flakiness

```
it('finds the word "bugs" on a card item', () ⇒ {
  cy,visit('/board/1')
  cy.get('[data-cy=card]')
    .eq(0)
    .should('contain.text', 'bugs')
})
```



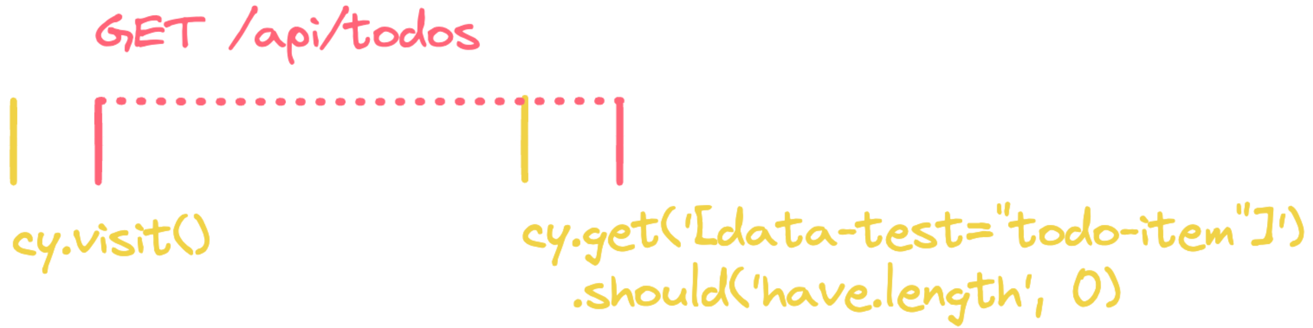
Network flakiness

```
it('add a todo item', () ⇒ {

  cy.visit('/')

  cy.get('[data-test="todo-item"]')
    .should('have.length', 0)

  cy.get('[data-test=new-todo]')
    .type('create code examples{enter}')

  cy.get('[data-test="todo-item"]')
    .should('have.length', 1)

});
```



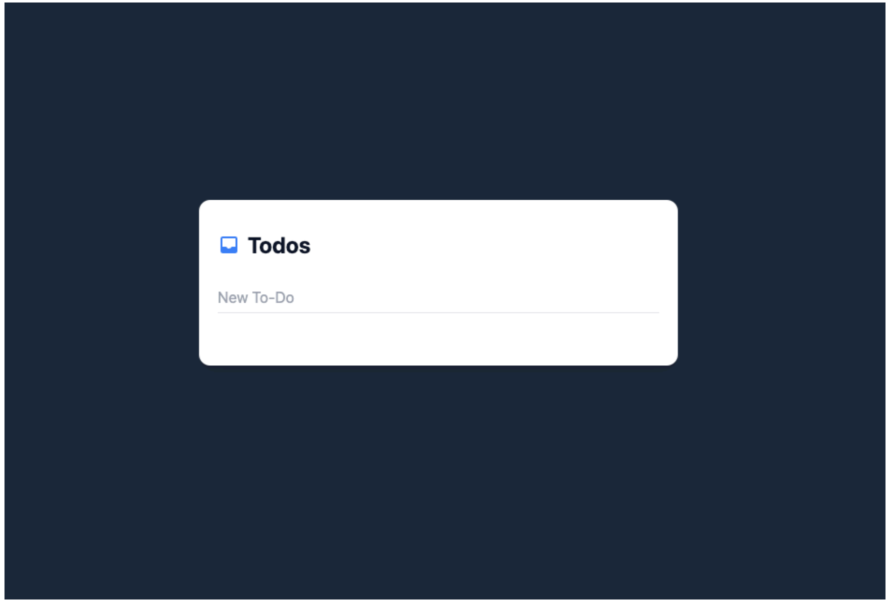
```
(xhr)  ● GET 200 /api/todos
(xhr)  ● POST 200 /api/todos
```

Network flakiness



Network flakiness

```
it('add a todo item', () ⇒ {

  cy.intercept('GET', '/api/todos')
    .as('todos')

  cy.visit('/')

  cy.wait('@todos')

  cy.get('[data-test="todo-item"]')
    .should('have.length', 0)

  cy.get('[data-test=new-todo]')
    .type('create code examples{enter}')

  cy.get('[data-test="todo-item"]')
    .should('have.length', 1)

});
```



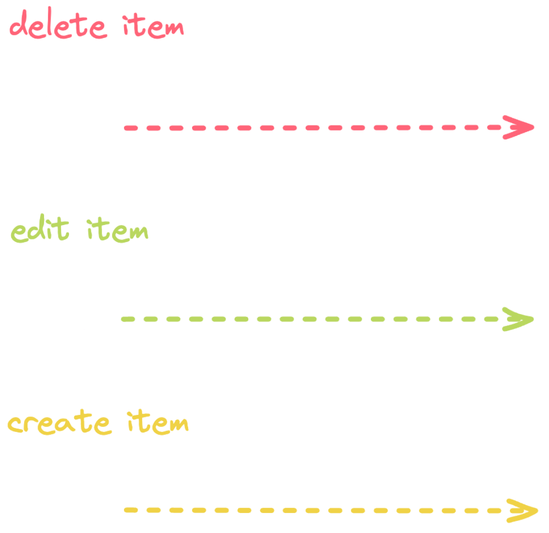
test execution
test coverage

Test isolation issues

❌ don't do this

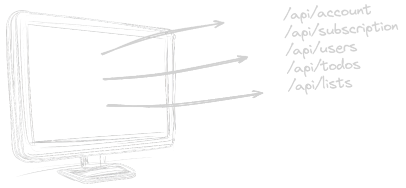
```
describe('item workflow', () => {

  it('creates an item', () => {

  })

  it('edits an item', () => {

  })

  it('deletes an item', () => {

  })
})
```

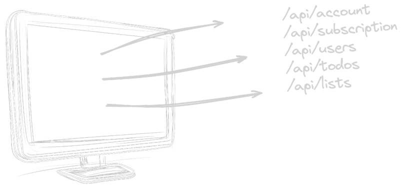
Test isolation issues

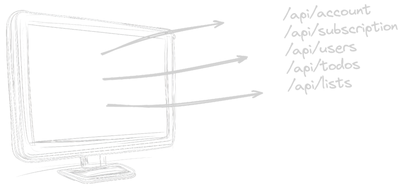


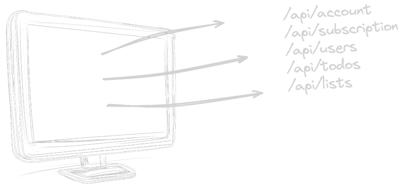
2 failed tests

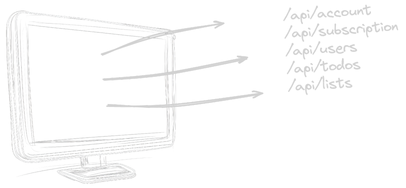
Infrastructure issues

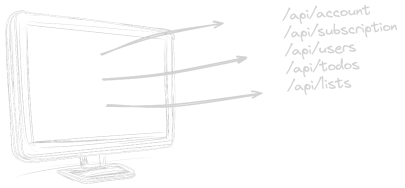


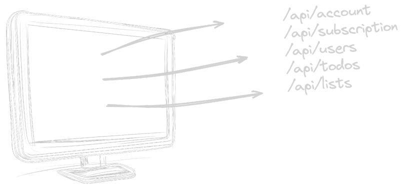


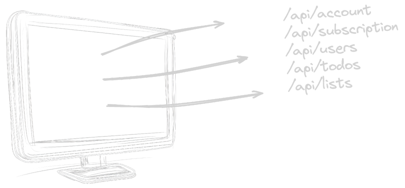












Infrastructure issues

```
beforeEach(() => {

  if (Cypress.currentRetry === 1) {
    Cypress.config('defaultCommandTimeout', 8000)
  }

  if (Cypress.currentRetry === 2) {
    Cypress.config('defaultCommandTimeout', 12000)
  }

})
```

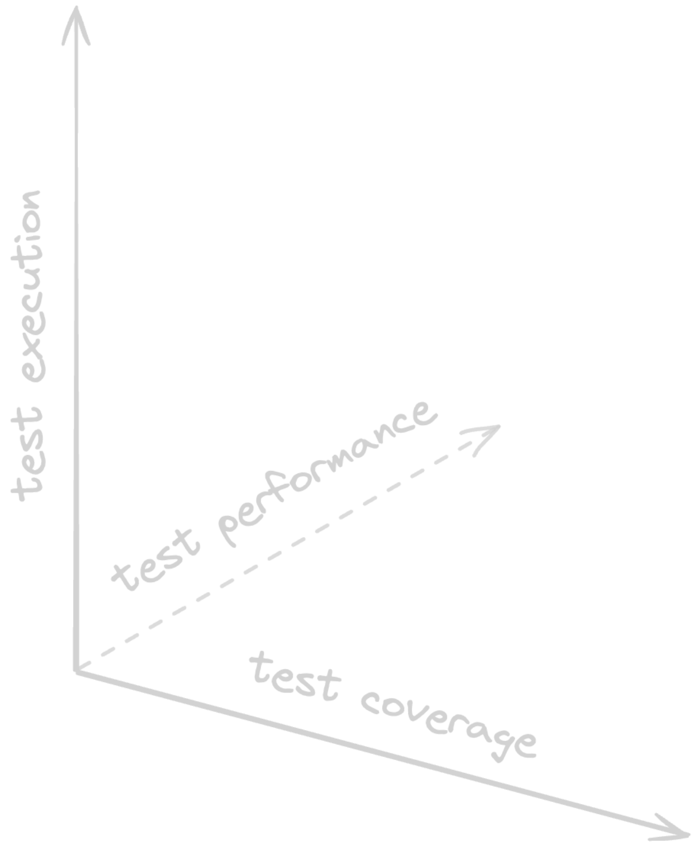
Which is better?

```
beforeEach(() ⇒ {})
before(() ⇒ {})
```

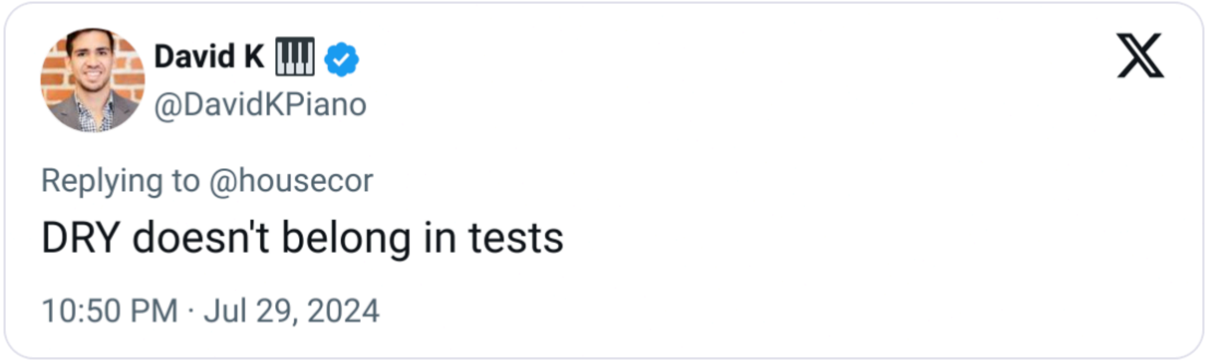
vs.

```
afterEach(() ⇒ {})
after(() ⇒ {})
```

opinion: cleanup should happen outside test suite


test execution
test performance
test coverage

DRY principle



David K 🎹
@DavidKPiano

Replying to @housecor

DRY doesn't belong in tests

10:50 PM · Jul 29, 2024

DRY principle

```
Cypress.Commands.add('login', () => {

  cy.visit('/')
  cy.get('#email').type('user@example.com')
  cy.get('#password').type('abcd1234')
  cy.get('#submit').click()
  cy.location('pathname').should('eq', '/home')

})
```

```
it('test #2', () => {

  cy.login() // go through UI

})

it('test #2', () => {

  cy.login() // go through UI

})

it('test #3', () => {

  cy.login() // go through UI

})
```

DRY principle

```
Cypress.Commands.add('login', () => {

  cy.session('user #1', () => {
    cy.visit('/')
    cy.get('#email').type('user@example.com')
    cy.get('#password').type('abcd1234')
    cy.get('#submit').click()
    cy.location('pathname').should('eq', '/home')
  })

})
```

```
it('test #2', () => {

  cy.login() // go through UI

})

it('test #2', () => {

  cy.login() // restore from cache

})

it('test #3', () => {

  cy.login() // restore from cache

})
```

Test idling

❌ incorrect way, don't use

```
cy.visit('/')
cy.wait(10000)
cy.get('button')
  .should('be.visible')
```

```
// cypress.config.js
import { defineConfig } from 'cypress'

export default defineConfig({
  e2e: {
    defaultCommandTimeout: 60000
  },
})
```

✅ much better

```
cy.visit('/')
cy.get('button', { timeout: 10000 })
  .should('be.visible')
```

```
// legacy part of the app that takes too long to load
it('test', { defaultCommandTimeout: 60000 }, () => {

})
```

Test idling

waiting for DOM elements

```
it('testing todos', () ⇒ {
  cy.visit('/')
  // page loading, wait for loader to disappear
  cy.get('.loader')
    .should('not.exist')

  // continue with our test
})
```

waiting for network calls

```
it('testing todos', () ⇒ {
  cy.intercept('/todos').as('todos')
  cy.visit('/')
  cy.wait('@todos')
  // page is loaded, continue with the test
})
```