

# Fast E2E Testing Using Cypress For Free

Gleb Bahmutov



**HUSTEF**

HUNGARIAN SOFTWARE TESTING FORUM



# Speaker: Gleb Bahmutov PhD

@bahmutov

gleb.dev

github.com/bahmutov

glebbahmutov.com/blog

www.youtube.com/glebbahmutov

  350.org   citizensclimatelobby.org  

Gleb Bahmutov

Sr Director of Engineering

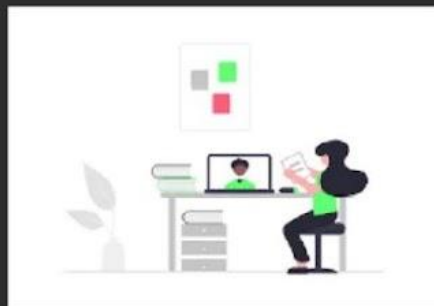
C / C++ / C# / Java /  
CoffeeScript / JavaScript / Node  
/ Angular / Vue / Cycle.js /  
functional programming /  
testing

# MERCARÍ

## Cypress Network Testing Exercises

advanced 

Practice writing advanced Cypress tests using `cy.intercept`, `cy.spy`, and other commands to solve each exercise. There are 100+ hands-on tests to write, each lesson comes with the full solution and a video explaining the solution and its principles step-by-step. Includes hours of videos showing the full solution to each exercise. The full course should take at least several weeks to finish.



Course price is \$149, group discounts are [available](#)

published March 2022

[Enroll in this course](#)

Looking to convince your company to buy this course? Send them [this link](#).  
Conducting test automation interviews covering this topic? Purchase my [interview kit](#) based on this course.

## Cypress Plugins

advanced 

Learn how to extend Cypress via its many many plugins. This course covers how a typical plugin works, how to write your own plugins, and the most popular plugins that many Cypress users include in their projects. Check out the the first lessons to get the taste of this advanced course.




Course price is \$149, group discounts are [available](#)

published July 2022

[Enroll in this course](#)

## Testing The Swag Store

beginner, e2e, component 

Learn how to write Cypress end-to-end and component tests for a web store. We will write complete feature tests, learn how to instantly log in, run tests on each commit on CI, organize code using TypeScript and Page Objects, write a few component tests, and achieve 100% code coverage. This is a good introduction to using Cypress for anyone who wants to add this powerful testing tool to their skillset.



Course price is \$99, group discounts are [available](#)

published February 2023

[Enroll in this course](#)

## Cypress vs Playwright

beginner, e2e 

Cypress and Playwright are two of the most popular tools for writing end-to-end web application tests. In this course you will learn both tools using several example applications. I will show where each tool shines and provides the best solution to your testing problems.



Course price is \$109, group discounts are [available](#)

started April 2023

[Enroll in this course](#)

<https://cypress.tips/courses>

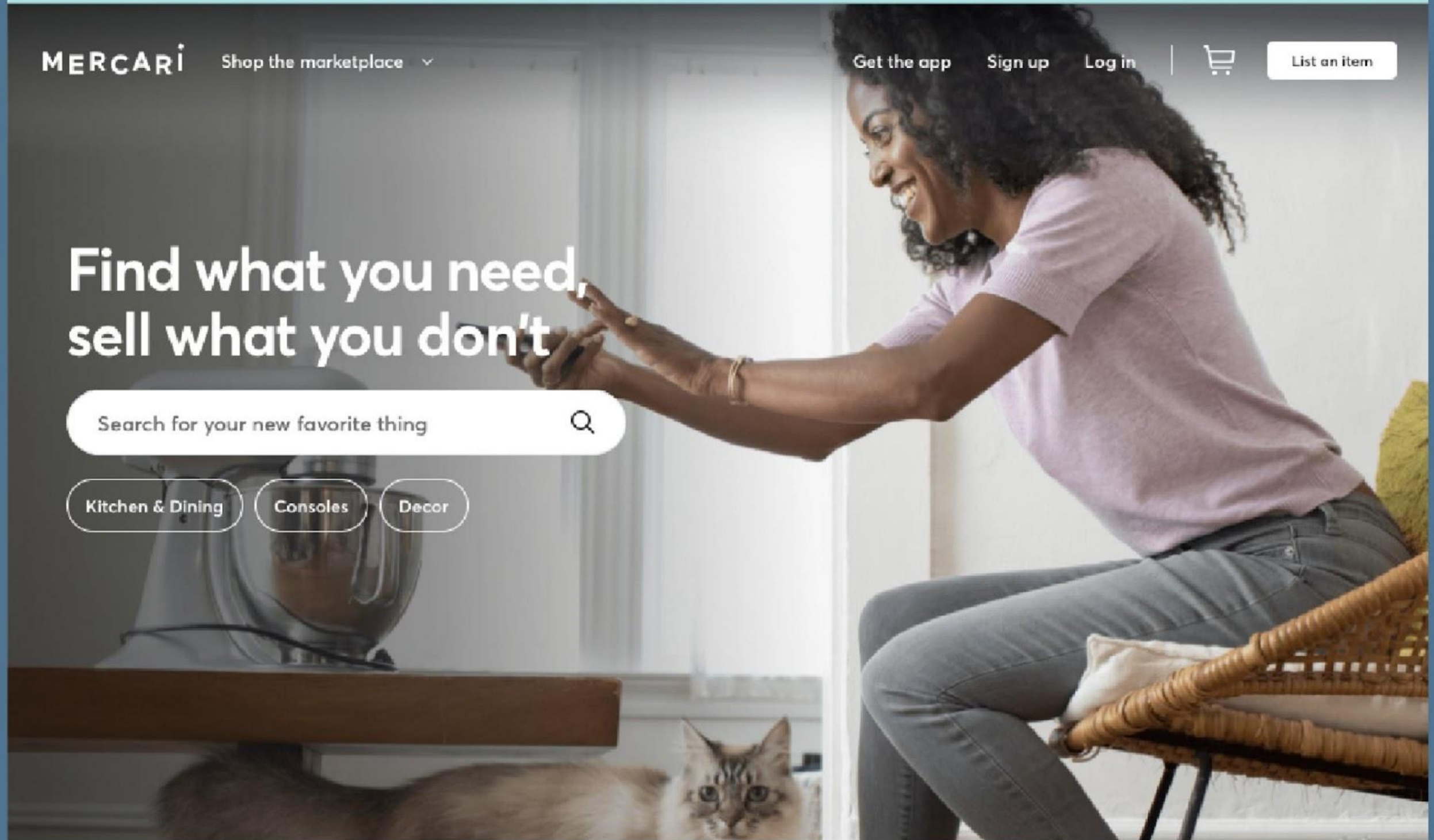
# Find what you need, sell what you don't

Search for your new favorite thing

Kitchen & Dining

Consoles

Decor



### Declutter

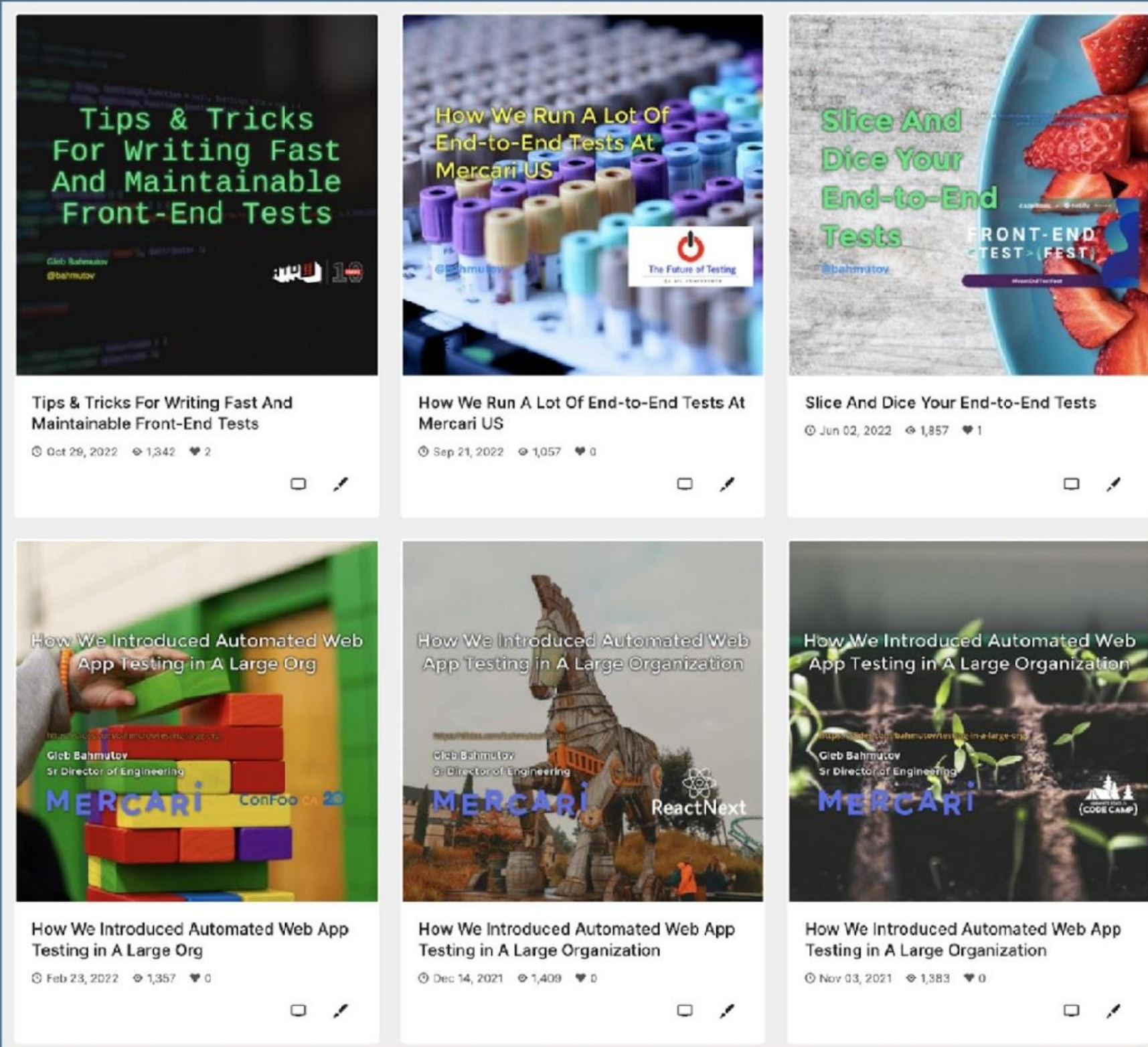
Breathe new life into your home and sell last year's go-tos.

### Discover

Shop one-of-a-kind treasures and all of your favorite brands.

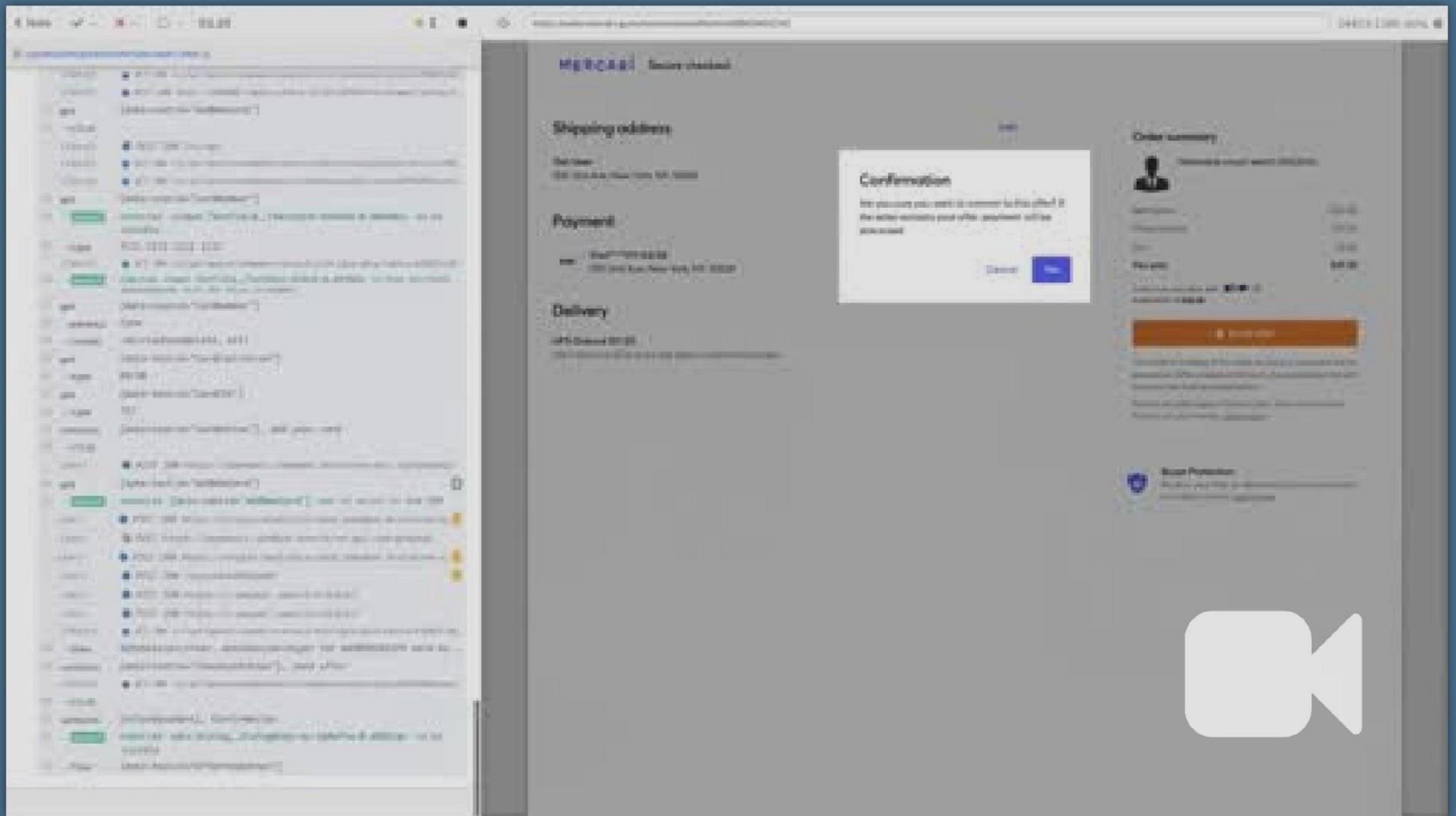
### All from home

Make your space feel more like home without ever leaving it.



# Mercari Does A Lot Of Testing

<https://slides.com/bahmutov/decks/mercari>



A typical Mercari US Cypress E2E test

## Test suite size

FILTER BY

main

Tag

Last 12 Months

Weekly

Reset Filters

### Test suite size over time



> 1000 E2E tests

Plus internal web application E2E tests



Why E2E Tests  
when we have  
unit tests?





It Is  
A Question  
Of Scale

## Unit test

```
1 expect(formatTime({ seconds: 3 }))  
2   .to.equal('00:03')
```

- Small chunks of code like functions and classes

## Component test

```
1 import { Component } from './Component'  
2 cy.mount(<Component props=... />)  
3 cy.get(...).click()
```

- Front-end React / Angular / Vue / X components
- Easy to test edge conditions

## End-to-End test

```
1 cy.visit('/')  
2 cy.get(...).click()
```

- Web application
- Easy to test the entire user flow

# Running Lots of E2E Tests: Problems

- Slow
- Flake
- Expensive

Cypress Cloud

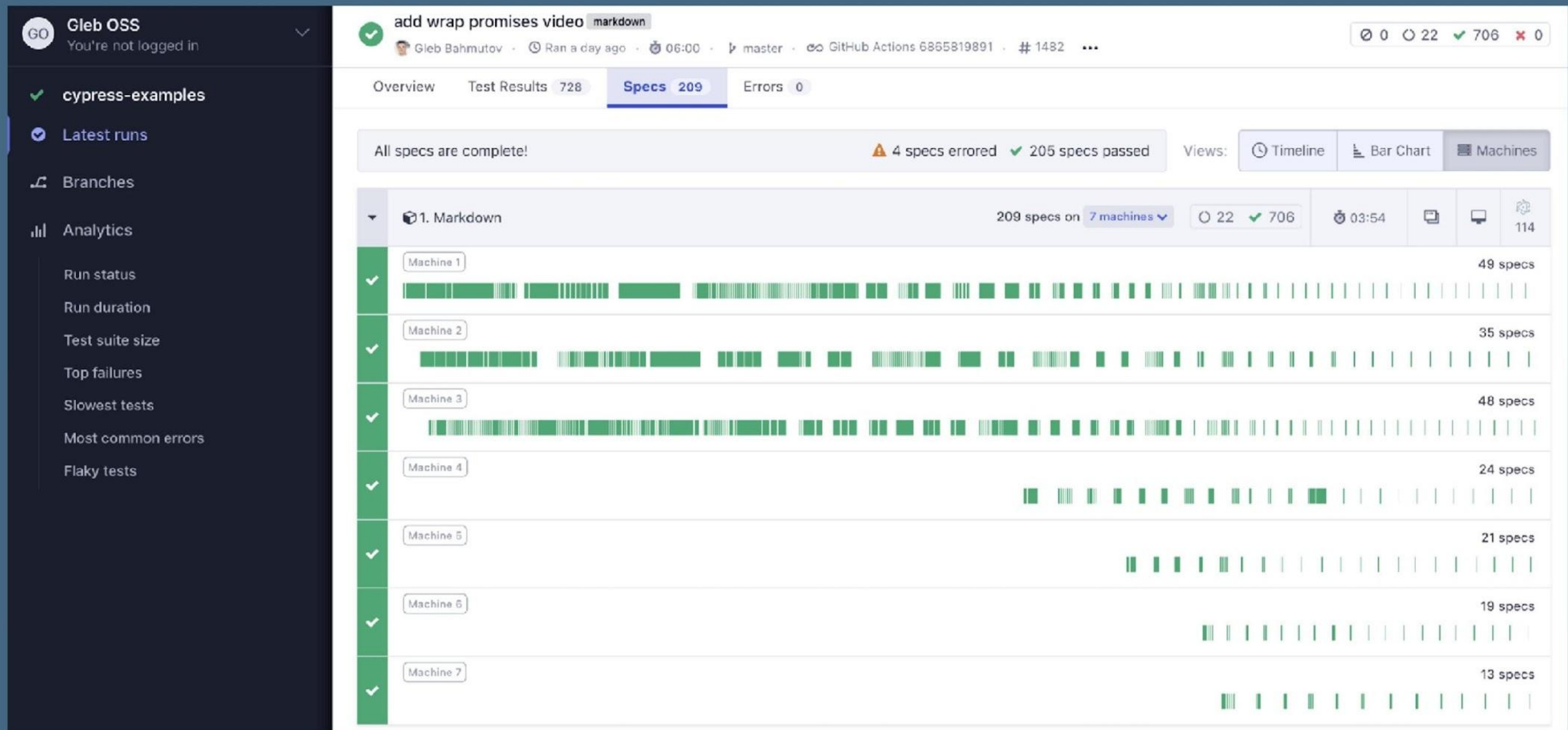
# Cypress Dashboard Cloud

The screenshot displays the Cypress Dashboard Cloud interface. On the left is a dark sidebar with navigation options: 'Gleb OSS' (with a 'GO' button and 'You're not logged in'), 'cypress-examples', 'Latest runs' (selected), 'Branches', and 'Analytics'. Under 'Analytics', there are links for 'Run status', 'Run duration', 'Test suite size', 'Top failures', 'Slowest tests', 'Most common errors', and 'Flaky tests'. The main content area is titled 'Latest runs' and includes a 'FILTER BY' section with dropdowns for 'All Time', 'Status', 'Branch', 'Committer', 'Tag', and 'Flaky Tests'. A 'Last updated: 10:21:36am' timestamp is shown. Below the filters is a list of five test runs, each with a green checkmark, a 'markdown' label, and a summary of results (0 failures, 22 or 23 passes, 702 or 704 total tests, 0 failures).

Run Title	Committer	Time	Branch	Provider	Run ID	Failures	Passes	Total	Failures	
add wrap promises video	Gleb Bahmutov	Ran a day ago	06:00	master	GitHub Actions 6865819891	# 1482	0	22	706	0
add link to the recipe	Gleb Bahmutov	Ran 2 days ago	04:31	master	GitHub Actions 6853356821	# 1481	0	22	704	0
add link to the video	Gleb Bahmutov	Ran 2 days ago	05:40	master	GitHub Actions 6853082897	# 1480	0	22	704	0
Update dependency cypress to v13.5.0	renovate[bot]	Ran 4 days ago	05:58	master	GitHub Actions 6833402603	# 1479	0	23	702	0
Update dependency cypress to v13.5.0	renovate[bot]	Ran 4 days ago	05:36	renovate/cypress-13.x	GitHub Actions 6833353279	# 1478	0	23	702	0

```
cypress run --record
```

# Cypress Dashboard Cloud



```
cypress run --record --parallel
```

# Cypress Dashboard Cloud

- test stats
- screenshots
- videos
- traces

The screenshot displays the Cypress Dashboard Cloud interface for a specific test run. At the top, it shows the run ID #1506, branch master, and file viewport.md. The test title is 'Viewport cy.viewport() - set the viewport size and dimension'. The test body is expanded to show the following steps:

- 1 `assert` initial width: expected 600 to be above 100
- 2 `log` initial, 600
- 3 `viewport` ipad-2
- 4 `log` still the same, 600
- 5 `viewport` iphone-4
- 6 `assert` after using viewport: expected 600 to equal ==600==
- 7 `viewport` 1200, 400
- 8 `log` current width, 1200
- 9 `window`
- 10 `its` .innerWidth
- 11 `- assert` expected 1200 to equal ==1200==

The central pane shows a browser view of the test code:

```
cy.viewport() -  
set the  
viewport size  
and  
dimension  
Test code  
  
// Cypress.config('viewportWidth')  
// cy.viewport() does not change the width  
const w = Cypress.config('viewportWidth')  
expect(w, 'initial width').to.be.greaterThan(100)  
cy.log('initial', w) // whatever the initial width is  
cy.viewport('ipad-2')  
cy.log('still the same', Cypress.config('viewportWidth'))
```

The right sidebar shows the network tab with four requests:

- 200 GET github.min.css
- 200 GET bootstrap.min.css
- 200 GET jquery-3.5.0.min.js
- 200 GET highlight.min.js

At the bottom, the progress bar indicates 'Attempt 1 of 1' and a duration of 108ms / 155ms.

cypress.io/pricing

Product Docs Community Company Pricing Log In Install >

**TRY CYPRESS CLOUD**

### Free

For small teams ready to get started with Cypress Cloud

Starting at  
**\$0** / month  
No credit card required

Users	50
Included test results	500
Additional results	-

What's included:

- ✓ Parallelization ?
- ✓ Load Balancing ?
- ✓ Test Replay ?
- ✓ Analytics ?
- ✓ Integrations ?

[Start for free](#)

### Team

For growing teams investing more in test automation

Starting at  
**\$67** / month  
Billed annually at \$799

Users	50
Included test results	120,000
Additional results	<a href="#">Calculate</a>

Everything in Free, plus:

- ✓ Flake Detection, History, & Severity ?
- ✓ Flaky Test Analytics ?
- ✓ Jira Integration ?
- ✓ Email Support ?

[Get started](#)

### Business

For businesses improving their quality and efficiency at scale

Starting at  
**\$267** / month  
Billed annually at \$3,199

Users	50
Included test results	120,000
Additional results	<a href="#">Calculate</a>

Everything in Team, plus:

- ✓ Spec Prioritization ?
- ✓ Auto Cancellation ?
- ✓ GitHub Enterprise & On-Prem GitLab ?
- ✓ SSO ?

[Get started](#)

### Enterprise

For organizations with custom needs at an enterprise scale

–

## Let's chat

[Start a conversation](#)

Users	Unlimited
Included test results	Custom
Additional results	<a href="#">Calculate</a>

Everything in Business, plus:

- ✓ Premium Support ?
- ✓ Custom Legal Terms ?
- ✓ Technical Account Manager ?
- ✓ Best Practices Sessions ?

[Contact us](#)

every it(...) executed = test result

10 tests x 1 per day = 300 test results

10 tests x 2 per day = 600 test results 🍅



***// It is really nice, if you  
can afford it***



# This Presentation

- How to run Cypress E2E specs in parallel for free 🎉
- How to run changed specs for free 🎉
- How to run tagged tests for free 🎉

Example repo: <https://github.com/bahmutov/fast-free>

1	Spec	Tests	Passing	Failing	Pending	Skipped
2	✓ spec-a.cy.js	1	1	-	-	-
3	✓ spec-b.cy.js	1	1	-	-	-
4	✓ spec-c.cy.js	1	1	-	-	-
5	✓ All specs passed!	3	3	-	-	-

```
6 it('works B', () => {
7   cy.wait(60_000)
8 })
9 // spec-c.cy.js
10 it('works C', () => {
11   cy.wait(10_000)
12 })
```

\$ npx cypress run

3 specs: 10 seconds, 1 minute, 10 seconds

← ci

✓ ci #1 Re-run all jobs ⋮

🏠 Summary

Jobs

✓ e2e

Run details

🕒 Usage

📄 Workflow file

**e2e** succeeded now in 2m 29s 🔍 Search logs ⚙️

>	✓ Set up job	1s
>	✓ Run actions/checkout@v4	1s
>	✓ Run cypress-io/github-action@v6	2m 26s
>	✓ Post Run actions/checkout@v4	0s
>	✓ Complete job	0s

Workflow run UI (github actions)

Summary
e2e
Search logs
⚙️

Jobs

- ✔️ e2e

Run details

- Usage
- Workflow file

Run cypress-io/github-action@v6 2m 26s

```

107 (Results)
108
109
110 | Tests:      1
111 | Passing:    1
112 | Failing:    0
113 | Pending:    0
114 | Skipped:    0
115 | Screenshots: 0
116 | Video:      false
117 | Duration:   10 seconds
118 | Spec Ran:   spec-c.cy.js
119
120
121
122 =====
123
124 (Run Finished)
125
126
127      Spec                                Tests  Passing  Failing  Pending  Skipped
128
129 | ✓ spec-a.cy.js                          00:10    1        1        -        -        -
130 |-----|-----|-----|-----|-----|
131 | ✓ spec-b.cy.js                          01:00    1        1        -        -        -
132 |-----|-----|-----|-----|-----|
133 | ✓ spec-c.cy.js                          00:10    1        1        -        -        -
134 |-----|-----|-----|-----|-----|
135 | ✓ All specs passed!                     01:20    3        3        -        -        -
136

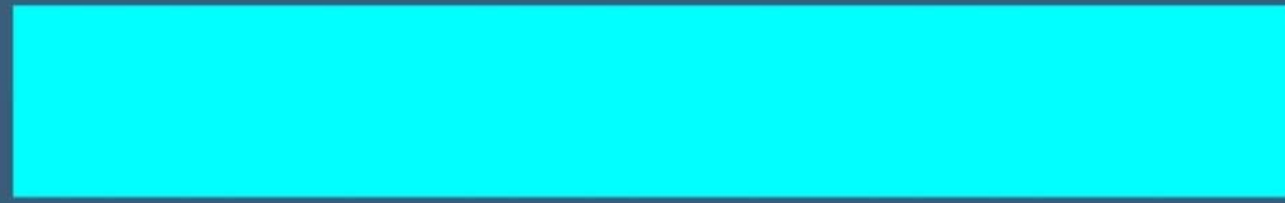
```

## Cypress GH Action details

# Waiting for 1 CI machine...



spec-a



spec-b



spec-c

*// Can't we use 2 CI machines?*

# Split Specs Manually

```
1 name: ci
2 on: push
3 jobs:
4   e2e-1:
5     runs-on: ubuntu-22.04
6     steps:
7       - uses: actions/checkout@v4
8       - uses: cypress-io/github-action@v6
9       with:
10        spec: cypress/e2e/spec-a.cy.js,cypress/e2e/spec-b.cy.js
11   e2e-2:
12     runs-on: ubuntu-22.04
13     steps:
14       - uses: actions/checkout@v4
15       - uses: cypress-io/github-action@v6
16       with:
17        spec: cypress/e2e/spec-c.cy.js
```

# Split Specs Manually

2 parallel containers

**ci.yml**  
on: push

Jobs

- ✓ e2e-1
- ✓ e2e-2

Run details

- Usage
- Workflow file

**e2e-1 summary**

### Cypress Results

Result	Passed ✓	Failed ✗	Pending 🤚	Skipped 🚫	Duration 🕒
Passing ✓	2	0	0	0	70.095s

Job summary generated at run-time

**e2e-2 summary**

### Cypress Results

Result	Passed ✓	Failed ✗	Pending 🤚	Skipped 🚫	Duration 🕒
Passing ✓	1	0	0	0	10.051s

# cypress-split plugin

```
1 name: ci
2 on: push
3 jobs:
4   e2e-1:
5     runs-on: ubuntu-22.04
6     steps:
7       - uses: actions/checkout@v4
8       - uses: cypress-io/github-action@v6
9       with:
10         spec: cypress/e2e/spec-a.cy.js,cypress/e2e/spec-b.cy.js
11   e2e-2:
12     runs-on: ubuntu-22.04
13     steps:
14       - uses: actions/checkout@v4
15       - uses: cypress-io/github-action@v6
16       with:
17         spec: cypress/e2e/spec-c.cy.js
```

automatically compute the spec lists for each machine



```
npm i -D cypress-split
```

```
1  const { defineConfig } = require('cypress')
2  // https://github.com/bahmutov/cypress-split
3  const cypressSplit = require('cypress-split')
4  module.exports = defineConfig({
5    e2e: {
6      setupNodeEvents(on, config) {
7        cypressSplit(on, config) ←
8        // IMPORTANT: return the config object
9        return config
10     },
11   },
12 })
```

cypress.config.js

# GitHub Actions Workflow

```
1 name: ci
2 on: push
3 jobs:
4   e2e-1:
5     runs-on: ubuntu-22.04
6     steps:
7       - uses: actions/checkout@v4
8       - uses: cypress-io/github-action@v6
9     env:
10       SPLIT: 2
11       SPLIT_INDEX: 0
12   e2e-2:
13     runs-on: ubuntu-22.04
14     steps:
15       - uses: actions/checkout@v4
16       - uses: cypress-io/github-action@v6
17     env:
18       SPLIT: 2
19       SPLIT_INDEX: 1
```

# Works On Other CIs

```
1 # GitLab CI
2 test:
3   stage: test
4   parallel: 2
5   script:
6     - npx cypress run --env split=true
```

```
1 # CircleCI
2 parallelism: 2
3 command: npx cypress run --env split=true
```

```
1 # many other CIs
2 # using process OS environment variables
3 job1: SPLIT=2 SPLIT_INDEX=0 npx cypress run
4 job2: SPLIT=2 SPLIT_INDEX=1 npx cypress run
```

<https://github.com/bahmutov/cypress-split#readme>

# cypress-split algorithm

- find all Cypress specs
- split into N chunks
- take chunk k

<https://github.com/bahmutov/find-cypress-specs>

e2e-1 summary

## cypress-split: chunk 1 of 2 (2 specs)

K	Spec	Passed	Failed	Pending	Skipped	Duration
1	cypress/e2e/spec-a.cy.js	1	0	0	0	10.049 seconds
2	cypress/e2e/spec-b.cy.js	1	0	0	0	1 minute, 0.046 seconds

[bahmutov/cypress-split](#)

### Cypress Results

Result	Passed	Failed	Pending	Skipped	Duration
Passing	2	0	0	0	70.095s

Job summary generated at run-time

e2e-2 summary

## cypress-split: chunk 2 of 2 (1 spec)

K	Spec	Passed	Failed	Pending	Skipped	Duration
1	cypress/e2e/spec-c.cy.js	1	0	0	0	10.051 seconds

[bahmutov/cypress-split](#)

### Cypress Results

Result	Passed	Failed	Pending	Skipped	Duration
Passing	1	0	0	0	10.051s

Job summary generated at run-time

The first machine's summary

The second machine's summary

# The GHA Job Matrix

```
1 name: ci
2 on: push
3 jobs:
4   e2e:
5     runs-on: ubuntu-22.04
6     strategy:
7       fail-fast: false
8       matrix:
9         containers: [1, 2]
10    steps:
11      - uses: actions/checkout@v4
12      - uses: cypress-io/github-action@v6
13      env:
14        SPLIT: ${ strategy.job-total }
15        SPLIT_INDEX: ${ strategy.job-index }
```

# Matrix Jobs UI

The screenshot displays the GitHub Actions interface for a workflow named "workflow matrix #5". The workflow is shown as successful with a green checkmark. The left sidebar contains navigation options: "Summary" (selected), "Jobs" (listing "e2e (1)" and "e2e (2)"), "Run details", "Usage", and "Workflow file". The main content area shows the workflow was triggered via push 2 minutes ago by user "bahmutov" on the "main" branch. A table summarizes the run: Status is "Success", Total duration is "2m 4s", and there are no artifacts. Below this, the "ci.yml" file content is shown, specifically the "on: push" trigger. A "Matrix: e2e" section is highlighted, showing a list of jobs: "e2e (1)" with a duration of "1m 47s" and "e2e (2)" with a duration of "39s".

← ci

✓ workflow matrix #5

Summary

Jobs

- ✓ e2e (1)
- ✓ e2e (2)

Run details

- Usage
- Workflow file

Triggered via push 2 minutes ago

Status: **Success**

Total duration: **2m 4s**

Artifacts: —

Triggered by: bahmutov pushed - 0bb924c main

ci.yml

on: push

Matrix: e2e

- ✓ e2e (1) 1m 47s
- ✓ e2e (2) 39s

The specs are split the same way



**Want To Go  
Faster?**



## Split specs across 5 machines

```
1 name: ci
2 on: push
3 jobs:
4   e2e:
5     runs-on: ubuntu-22.04
6     strategy:
7       fail-fast: false
8       matrix:
9         containers: [1, 2, 3, 4, 5]
10    steps:
11      - uses: actions/checkout@v4
12      - uses: cypress-io/github-action@v6
13      env:
14        SPLIT: ${ strategy.job-total }
15        SPLIT_INDEX: ${ strategy.job-index }
```

## Split specs across 10 machines

```
1 name: ci
2 on: push
3 jobs:
4   e2e:
5     runs-on: ubuntu-22.04
6     strategy:
7       fail-fast: false
8       matrix:
9         containers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
10    steps:
11      - uses: actions/checkout@v4
12      - uses: cypress-io/github-action@v6
13      env:
14        SPLIT: ${strategy.job-total}
15        SPLIT_INDEX: ${strategy.job-index}
```

Tip: use [bahmutov/gh-build-matrix](#) to create strategy array for N machines

# About that split...



1



spec-a



spec-b



2



spec-c

*// Is this the best we can do?*

# About that split...



1



spec-b



2



spec-a



spec-c

We know the spec-b takes a lot longer

# Idea: use spec durations

run this command locally

```
$ SPLIT_FILE=timings.json SPLIT=1 SPLIT_INDEX=0 npx cypress run
```

```
1  {
2    "durations": [
3      {
4        "spec": "cypress/e2e/spec-a.cy.js",
5        "duration": 10073
6      },
7      {
8        "spec": "cypress/e2e/spec-b.cy.js",
9        "duration": 60090
10     },
11     {
12       "spec": "cypress/e2e/spec-c.cy.js",
13       "duration": 10075
14     }
15   ]
16 }
```

commit timings.json to the  
source code repo

# cypress-split timings algorithm

- find all Cypress specs
- look up timings
- sort specs by duration from longest to shortest
- fill N machine chunks
  - put each spec into the chunk with smallest sum of durations



spec-a



spec-b



spec-c



spec-b



spec-a



spec-c



spec-b



spec-a



spec-c

# Update timings.json nightly

```
1 name: nightly
2 on:
3   # run this workflow every night at 3am
4   schedule:
5     - cron: '0 3 * * *'
6   # or when the user triggers it from GitHub Actions page
7   workflow_dispatch:
8 jobs:
9   e2e:
10    runs-on: ubuntu-22.04
11    steps:
12      - uses: actions/checkout@v4
13      - uses: cypress-io/github-action@v6
14      with:
15        # we don't need Cypress own runner summary
16        publish-summary: false
17    env:
```

[.github/workflows/nightly.yml](#)



Run cypress-io/github-action@v6

```
135
136 cypress-split: here are passing spec timings
137 {
138   "durations": [
139     {
140       "spec": "cypress/e2e/spec-b.cy.js",
141       "duration": 60118
142     },
143     {
144       "spec": "cypress/e2e/spec-c.cy.js",
145       "duration": 10059
146     },
147     {
148       "spec": "cypress/e2e/spec-a.cy.js",
149       "duration": 10063
150     }
151   ]
152 }
```

153 cypress-split: spec timings unchanged

154

155 =====

156

(Run Fin

158

159

160

161

162

163

164

165

166

167

168

169

Commit changed spec timings 🕒

1 ▶ Run stefanzweifel/git-auto-commit-action@v4

21 Started: bash /home/runner/work/\_actions/stefanzweifel/git-auto-commit-action/v4/entrypoint.sh

22 INPUT\_REPOSITORY value: .

23 INPUT\_STATUS\_OPTIONS:

24 INPUT\_FILE\_PATTERN: timings.json

25 Working tree clean. Nothing to commit.

GitHub Actions integration with  
GitHub = 🍊🍊🍊



**Make It Simple**

```
1 name: split
2 on:
3   workflow_dispatch:
4 jobs:
5   split:
6     uses: bahmutov/cypress-workflows/.github/workflows/split.yml@v2
7     with:
8       nE2E: 2
```

[.github/workflows/split.yml](https://github.com/bahmutov/cypress-workflows/.github/workflows/split.yml)  
<https://github.com/bahmutov/cypress-workflows>

The screenshot shows the GitHub Actions interface for a workflow run named 'split #1'. The workflow is triggered manually and has a status of 'Success' with a total duration of 2m 4s. The workflow file is 'split.yml' and it is triggered by 'workflow\_dispatch'. The workflow consists of several jobs:

- split / prepare**: 15s, successful.
- Matrix: split / e2eTests**: 2 jobs completed, successful. This job branches into two parallel jobs:
  - split / merge-split-timings**: 0s, pending.
  - split / merge-coverage**: 0s, pending.
- Matrix: split / componentTests**: 1 job completed, pending. This job branches into two parallel jobs:
  - split / merge-reports**: 0s, pending.
  - split / merge-split-timings**: 0s, pending.

The interface also includes a sidebar with navigation options: Summary, Jobs, Run details, Usage, and Workflow file.

```
1 name: split
2 on:
3   workflow_dispatch:
4 jobs:
5   split:
6     uses: bahmutov/cypress-workflows/.github/workflows/split.yml@v2
7     with:
8       nE2E: 2
9     split-file: 'timings.json'
```

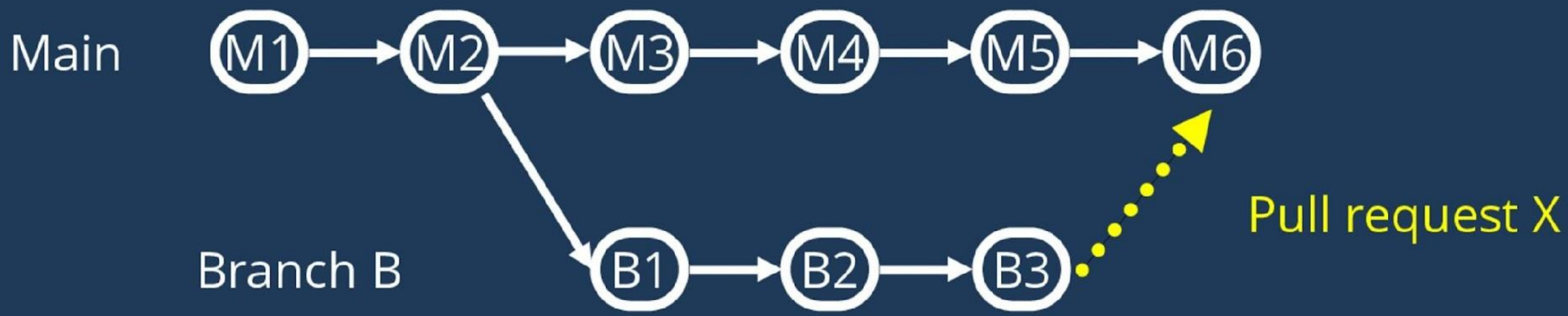


# This Presentation

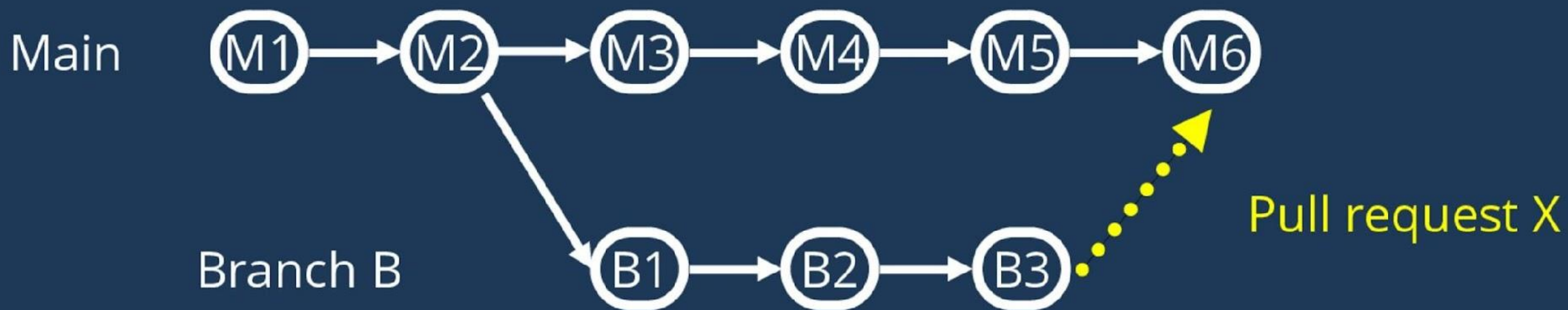
- **How to run Cypress E2E specs in parallel** for free 🎉
- How to run changed specs for free 🎉
- How to run tagged tests for free 🎉

# Which Tests To Run For This Pull Request?





what tests/specs should  
we run?



**Idea:** find and run specs that were changed in the branch B with respect to the parent commit M2

```
npx find-cypress-specs --branch main --parent  
spec-a.cy.js,spec-c.cy.js
```



# Run the changed specs

```
npx cypress run --spec spec-a.cy.js,spec-c.cy.js
```

we can still split these specs  
across N machines!

# If the changed specs pass, run all tests

Cancel workflow



## Summary

### Jobs

- ✅ find-changed-specs
- 🕒 changed-specs
  - ✅ prepare
  - 🕒 e2eTests (1)
  - 🕒 e2eTests (2)
  - 🕒 e2eTests (3)
  - 🕒 componentTests

### Run details

- 🕒 Usage
- 📄 Workflow file

Triggered via pull request 3 minutes ago

Status	Total duration	Artifacts
<b>In progress</b>	-	-

### pr.yml

on: pull\_request

✅ find-changed-specs 2m 26s



### find-changed-specs summary

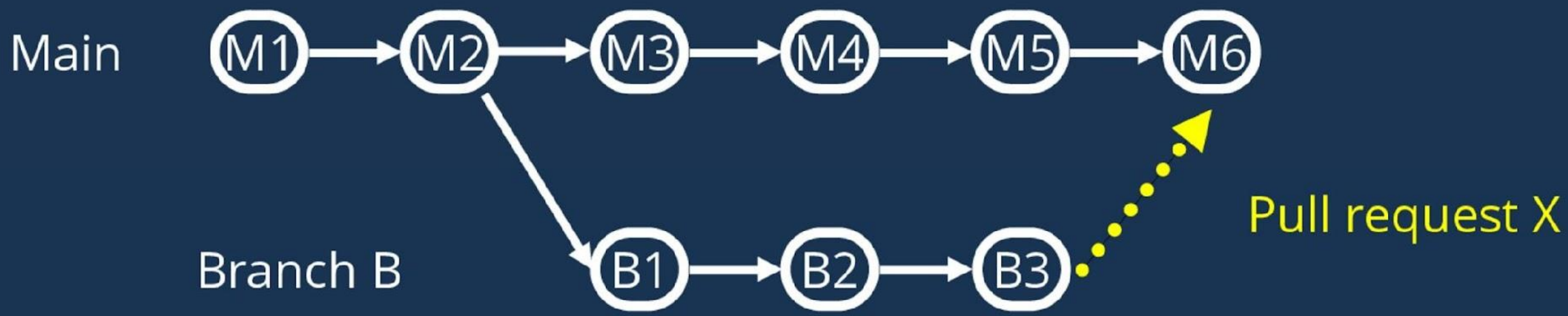
Found that 8 specs changed: cypress/e2e/adding-spec.ts, cypress/e2e/clear-completed-spec.ts, cypress/e2e/complete-all-spec.ts, cypress/e2e/editing-spec.ts, cypress/e2e/item-spec.ts, cypress/e2e/persistence-spec.ts, cypress/e2e/routing-spec.ts, cypress/e2e/spec.ts  
 estimated 3 machines needed



[Job summary generated at run-time](#)

# This Presentation


- How to run Cypress E2E specs in parallel for free 🎉
- How to run changed specs for free 🎉
- How to run ~~tagged~~ tests for free 🎉  
*the right*



pull request has specs  
and source file changes

```
1 <InputError
2   isError={Boolean(error)}
3   type={INPUT_TYPES.TEXT}
4   value={lastName}
5   onChange={handleLastNameChange}
6   testId="lastName"
7   placeholder="Last name"
8   // Custom
9   id="last-name"
10  autoComplete="off"
11  autoCapitalize="none"
12 />
```

test id attribute



a changed source file (React JSX)

```
1 // spec-b.cy.js
2 cy.get( '[data-testid=lastName]' ).should( 'be.visible' )
```

```
1 // spec-z.cy.js
2 cy.get( '[data-testid=lastName]' )
3   .should( 'not.exist' )
```

specs that check elements with the test id  
"lastName"

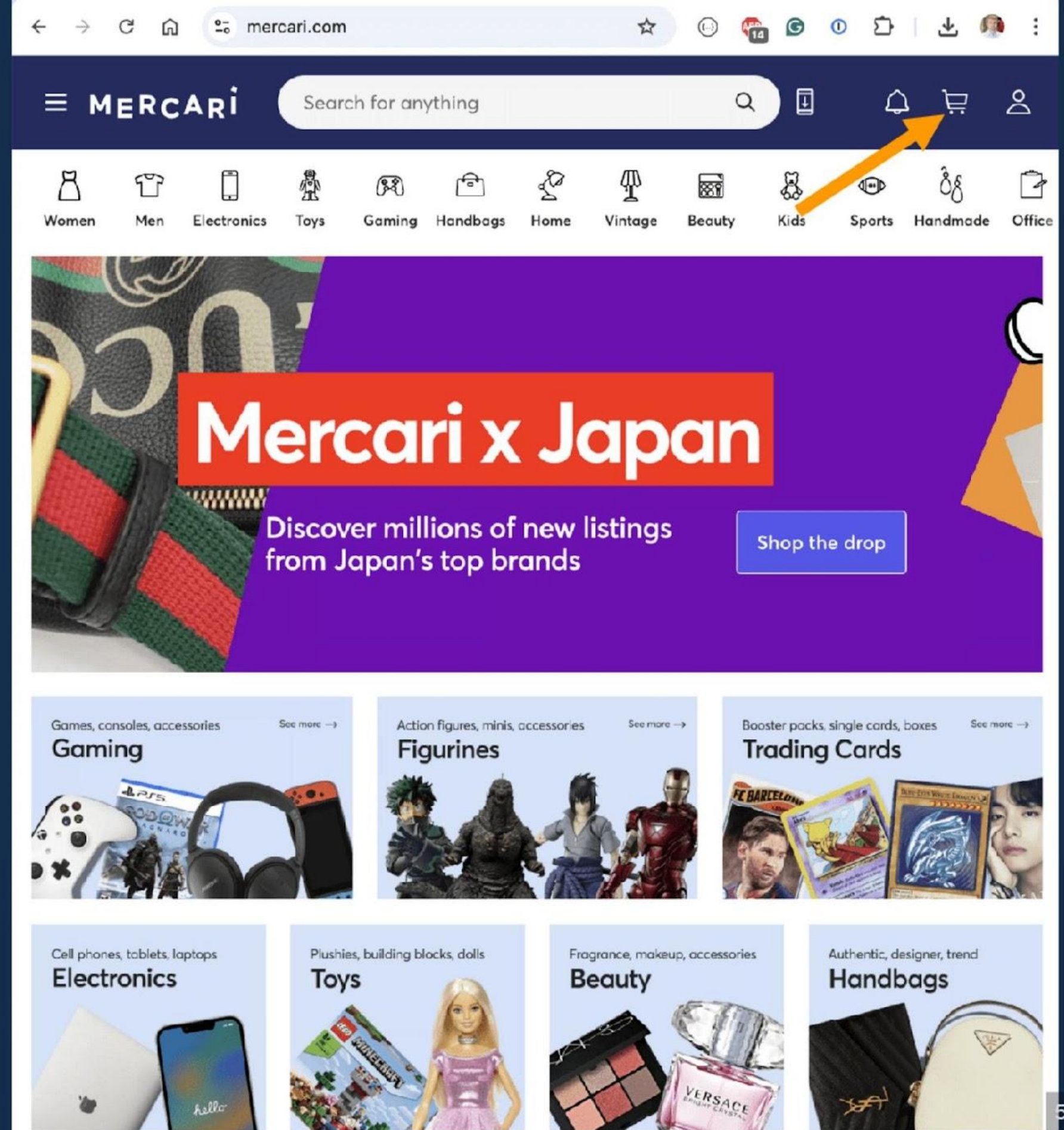
```
npx find-ids --test-ids lastName
spec-b.cy.js, spec-z.cy.js
```

Run the found specs!

# Specs based on changes:

- Find test ids in the changed source files
- Run specs that use these test ids
- (Optional) run the rest of the specs

Bonus:  
button does  
not work 😞





Look up its  
test id

Launch all  
E2E

tests that  
"touch"

this element

gleb.dev

The image shows a browser window displaying the Mercari website. The browser's address bar shows 'mercari.com'. The website header includes the Mercari logo, a search bar, and navigation icons for home, notifications, and cart. Below the header is a category menu with icons for Women, Men, Electronics, Toys, Gaming, Handbags, Home, Vintage, Beauty, Kids, Sports, Handmade, and Office. The main content area features a large banner for 'Mercari x Japan' with the text 'Discover millions of new listings from Japan's top brands' and a 'Shop the drop' button. The browser's developer tool is open at the bottom, showing the 'Elements' panel. The selected element is a `<div width="40px" data-testid="CartIconWithCounter" class="CartIconWithCounter_CartWrapper-sc-310ba1ba-0 bNlNaG"> == $0`. An orange arrow points from the text 'Look up its test id' to the cart icon in the browser. Another orange arrow points from the text 'this element' to the selected HTML element in the developer tool.

# Slice Tests With Tags



```
1 describe('Shipping', { tags: '@shipping' }, () => {
2   it(
3     'C1234 uses the default Mercari shipping',
4     { tags: ['@sanity', '@regression', '@mobile'] },
5     () => {
6       ...
7     }
8   )
9 })
```

Learn more:

<https://slides.com/bahmutov/slice-and-dice>

<https://glebbahmutov.com/blog/tag-tests/>

<https://github.com/bahmutov/find-cypress-specs>

```
1 describe('Shipping', { tags: '@shipping' }, () => {
2   it(
3     'C1234 uses the default Mercari shipping',
4     { tags: ['@sanity', '@regression', '@mobile'] },
5     () => {
6       ...      Effective tags
7     }          @shipping, @sanity
8   )           @regression, @mobile
9 })
```

Learn more:

<https://slides.com/bahmutov/slice-and-dice>

<https://glebbahmutov.com/blog/tag-tests/>

<https://github.com/bahmutov/find-cypress-specs>

```

1 describe('Shipping', { tags: '@shipping' }, () => {
2   it(
3     'C1234 uses the default Mercari shipping',
4     { tags: ['@sanity', '@regression', '@mobile'] },
5     () => {
6       ...      Effective tags
7     }          @shipping, @sanity
8   )           @regression, @mobile
9 })

```

```

1 $ find-cypress-specs --tags
2
3 Tag          Tests
4 -----
5 @admin       21
6 @analytics   7
7 @balance     14
8 @careers     2
9 @comments    33
10 @coupon      21
11 @cross-border 6
12 @email       41
13 @furniture   3
14 @guest       19
15 @helpcenter  99
16 @home        13
17 ...

```

Learn more:

<https://slides.com/bahmutov/slice-and-dice>

<https://glebbahmutov.com/blog/tag-tests/>

<https://github.com/bahmutov/find-cypress-specs>

@profile

@sell

@login

@shipping

@payment

@...

■ ■ ■ ■ ■ ■ ■ @sanity

■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■ @regression

■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■  
■ ■ ■ ■ ■ ■ ■ ■ ■ all



## Cypress tests to run

Any changed specs in this PR will run first, then you can pick which additional E2E tests to execute, see [this blog post](#).

- run tests tagged `@sanity` (default, a small set of tests)
- run tests tagged `@regression` (a larger set across multiple features)
- run all tests (overwrites all other options, runs all tests)

## Run tests by topics

Pick all topics that you want to test

- run tests tagged `@balance`
- run tests tagged `@helpcenter`
- run tests tagged `@local`
- run tests tagged `@login`
- run tests tagged `@mobile`
- run tests tagged `@offer`
- run tests tagged `@payment`
- run tests tagged `@profile`
- run tests tagged `@search`
- run tests tagged `@sell`
- run tests tagged `@shipping`
- run tests tagged `@signup`
- run tests tagged `@w9`
- run tests tagged `@careers`
- run tests tagged `@messaging`

To re-run the tests, pick the tags above then click the checkbox below

- re-run the tests

`@sanity,`  
`@regression, all`

`features`

+ changed specs  
+ changes by test ids

# Every Day 3x

# Mercai US Today

Mercai runs all E2E tests (in parallel using 20 machines) in 30 minutes

## When a PR is opened

- run all changed specs
- run all specs testing the source changes (by test id)
- run the test tags picked by dev / QA



Future: pick tests by an API call

Launch all E2E tests that "touch" this API call

The image shows a screenshot of the Mercari website. The top navigation bar includes the Mercari logo, a search bar, and icons for mobile, notifications, shopping cart, and user profile. Below the navigation bar is a category menu with icons for Women, Men, Electronics, Toys, Gaming, Handbags, Home, Vintage, Beauty, Kids, Sports, Handmade, Office, and Pets. The main content area features a large banner for "Mercari x Japan" with the text "Discover millions of new listings from Japan's top brands" and a "Shop the drop" button. At the bottom, a browser developer tool is open, showing the Network tab. The selected request is an API call to "/v1/api" with the operation name "homeQuery". The headers for this request include "X-Gql-Operation-Name: homeQuery" and "X-Ld-Variants: web-intro-apple-express-checkout:2;web-new-paypal-popup:0;us-web-checkout-payments-zip-banner:0;us-web-checkout-payments-zip-widget:2;search\_synonyms\_web:0;web\_merch".

MERCARI Search for anything

Women Men Electronics Toys Gaming Handbags Home Vintage Beauty Kids Sports Handmade Office Pe

Mercari x Japan

Discover millions of new listings from Japan's top brands

Shop the drop

Elements Recorder Console Sources Network >> 249 186

Preserve log Disable cache No throttling

/v1/api Invert Hide data URLs Hide extension URLs

All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other Blocked response cookies

Blocked requests 3rd-party requests

Name	Headers	Payload	Preview	Response	Initiator	Timing
api?operationName=TrackEvent						
api?operationName=homeQuery&variabl...	X-Gql-Operation-Name: homeQuery					
api?operationName=multiSearchQuery&...	X-Ld-Variants: web-intro-apple-express-checkout:2;web-new-paypal-popup:0;us-web-checkout-payments-zip-banner:0;us-web-checkout-payments-zip-					
api?operationName=TrackEvent&operati...						
api?operationName=TrackEvent						
api?operationName=TrackEvent						

65

👏 Thank you 👏

Slides at <https://slides.com/bahmutov/fast-and-free-e2e>

Gleb Bahmutov

gleb.dev



**HUSTEF**  
HUNGARIAN SOFTWARE TESTING FORUM

