

Running A Thousand End-To-End Cypress Tests Every Day

Gleb Bahmutov
gleb.dev





Climate Crisis Is Bad



SIERRA
CLUB



Citizens' Climate Lobby



**Environmental
Voter Project**

Join others to fight
the climate crisis

Speaker: Gleb Bahmutov PhD

 bahmutov.bsky.social

gleb.dev

github.com/bahmutov


glebbahmutov.com/blog

www.youtube.com/glebbahmutov

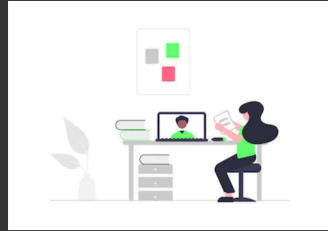
  350.org   citizensclimatelobby.org  

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

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

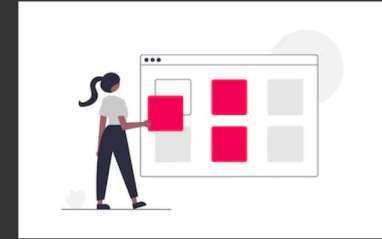
[Enroll in this course](#)

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](#)

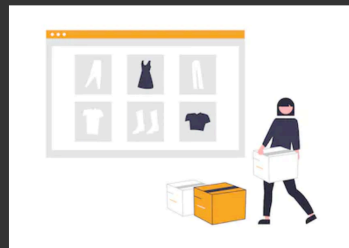
published July 2022

[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](#)

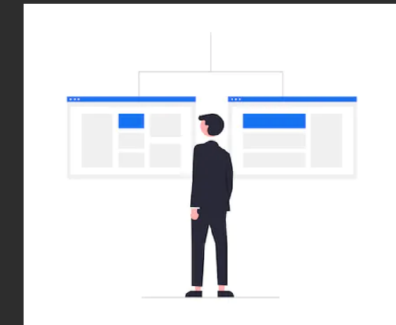
published February 2023

[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.

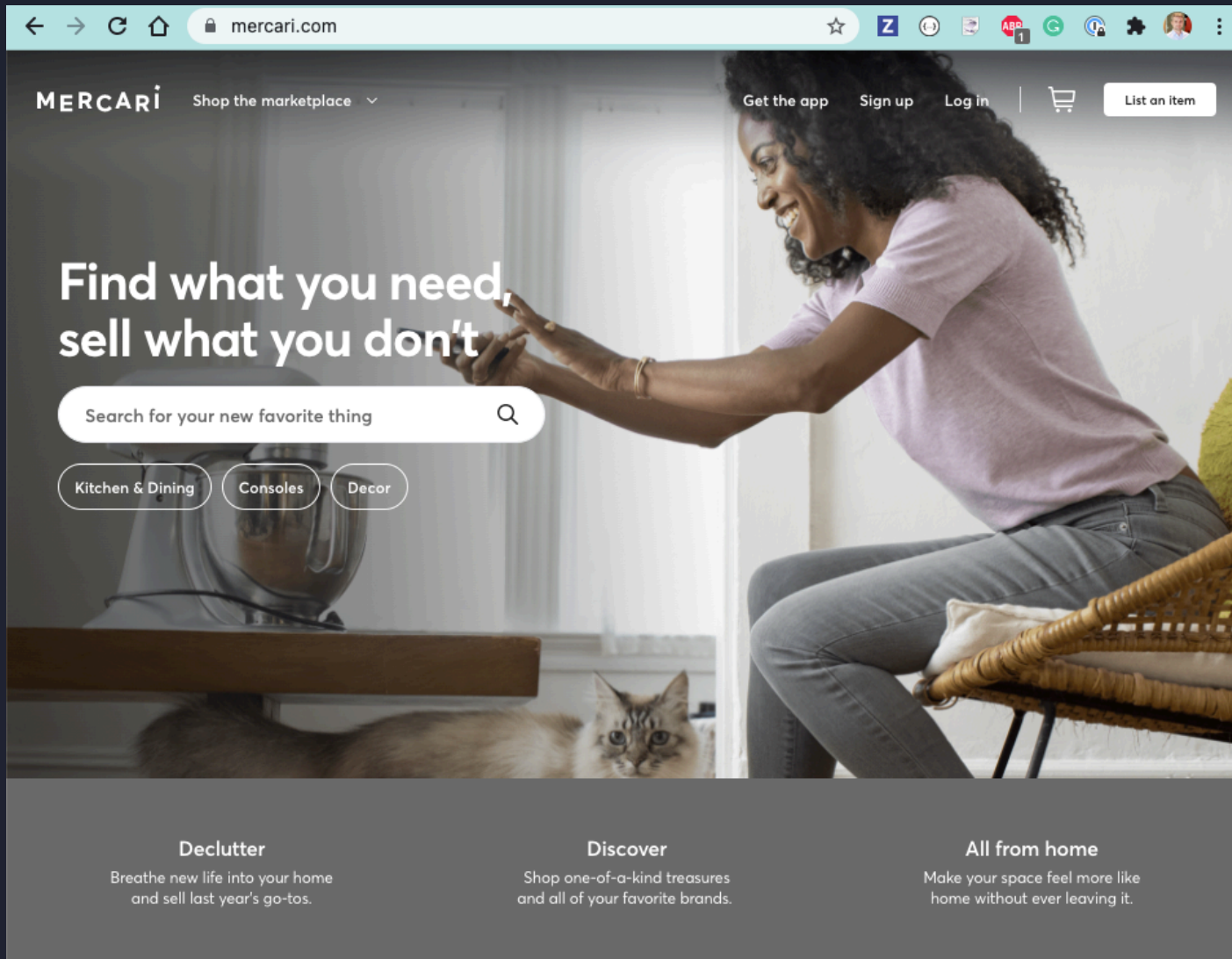


Course price is \$109, group discounts are [available](#)

started April 2023

[Enroll in this course](#)

<https://cypress.tips/courses>



Gleb Bahmutov
Sr Director of
Engineering

MERCARÍ

 cypress.io

MERCARÍ



EveryScape

75 → 20

MathWorks

2000

Kensho

8 → 100

Cypress.io

5 → 50

MERCARI

150 (2000)

“Gleb, we need our stuff to work.”

- Web app
- ReactNative mobile app
- APIs
- Special projects

Agenda

- Test speed
- Fast(er) CI feedback
- Test tags
- Everyone is involved
- Future is ☀️

<https://slides.com/bahmutov/1000-e2e>



How We Introduced Automated Web App Testing in A Large Org

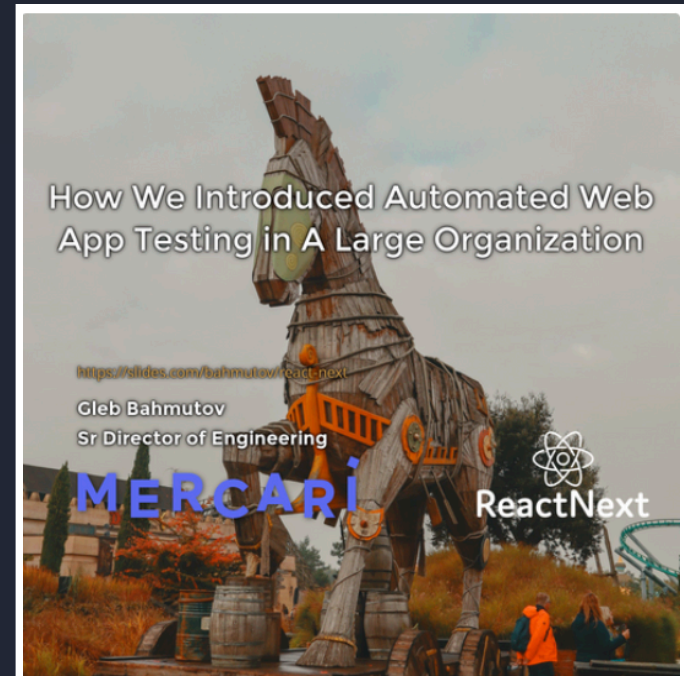
<https://slides.com/bahmutov/testing-large-org>

Gleb Bahmutov
Sr Director of Engineering

MERCARI ConFoo CA 20

How We Introduced Automated Web App Testing in A Large Org

🕒 Feb 23, 2022 👁 473 ❤️ 0



How We Introduced Automated Web App Testing in A Large Organization

<https://slides.com/bahmutov/react-next>

Gleb Bahmutov
Sr Director of Engineering

MERCARI ReactNext

How We Introduced Automated Web App Testing in A Large Organization

🕒 Dec 14, 2021 👁 546 ❤️ 0



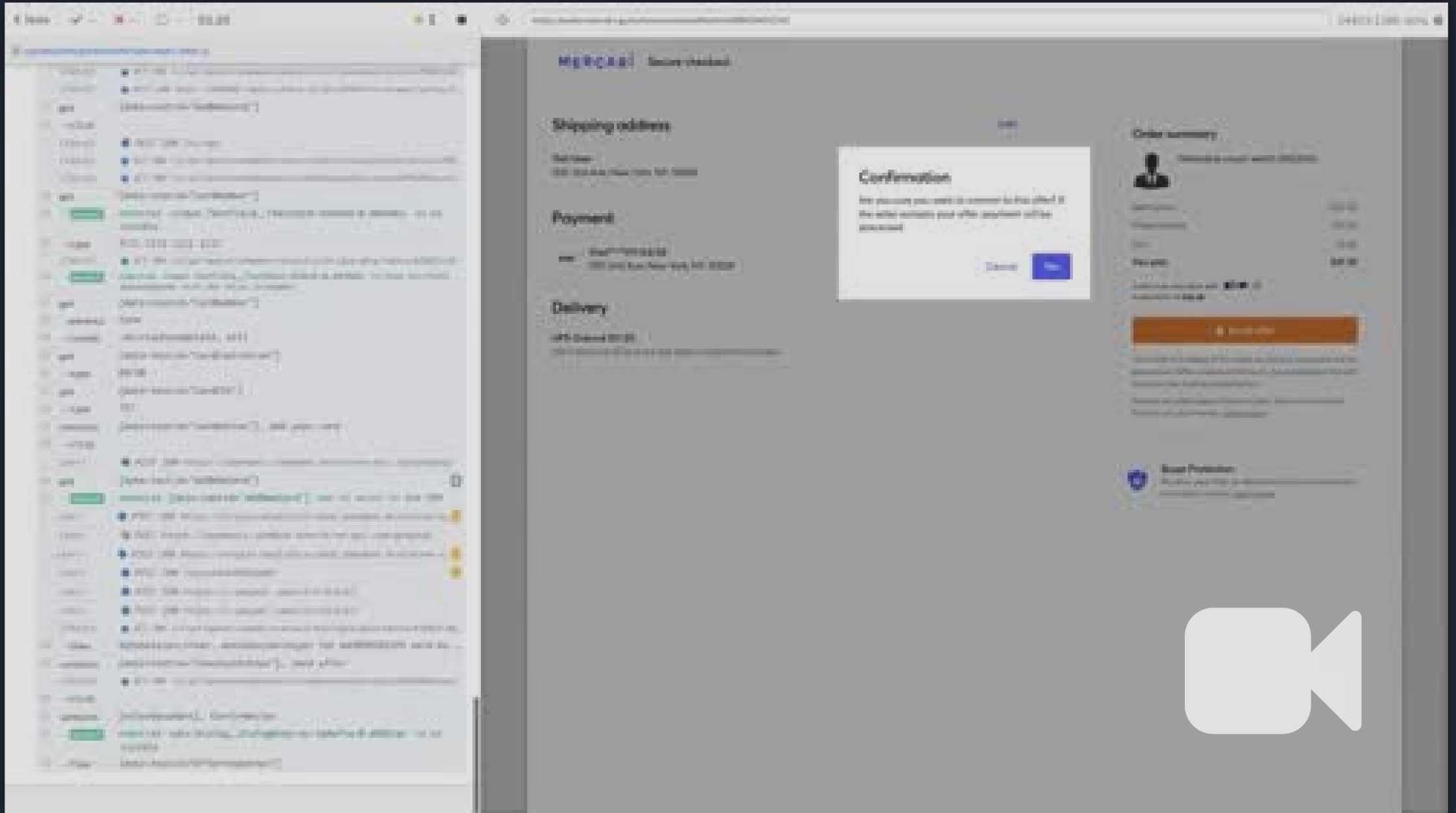
slides.com/bahmutov/testing-large-org

slides.com/bahmutov/react-next

slides.com/bahmutov/decks/mercari



*“ If you like it,
Put a  **test** on it...”*



A typical Mercari US Cypress E2E test

```
1 cy.signup(seller)
2
3 cy.createListing({
4   name: `Macbook one ${Cypress._.random(1e10)}`,
5   description: 'Seller will delete all items',
6   price: 198,
7 })
8 cy.createListing({
9   name: `Macbook two ${Cypress._.random(1e10)}`,
10  description: 'Seller will delete all items',
11  price: 199,
12 })
13
14 visitBlankPage()
15 cy.loginUserUsingAPI(seller)
16 cy.visitProtectedPage('/mypage/listings/active')
17 cy.byTestId('Filter', 'Active').should('be.visible').and('contain', '2')
18 cy.byTestId('ListingRow').should('be.visible').and('have.length', 2)
```

A typical Cypress test

```
1 cy.signup(seller)
2
3 cy.createListing({
4   name: `Macbook one ${Cypress._.random(1e10)}`,
5   description: 'Seller will delete all items',
6   price: 198,
7 })
8 cy.createListing({
9   name: `Macbook two ${Cypress._.random(1e10)}`,
10  description: 'Seller will delete all items',
11  price: 199,
12 })
13
14 visitBlankPage()
15 cy.loginUserUsingAPI(seller)
16 cy.visitProtectedPage('/mypage/listing')
17 cy.byTestId('Filter', 'Active').should('be.visible')
18 cy.byTestId('ListingRow').should('be.visible')
```

A typical Cypress test

```
PULL_REQUEST_TEMPLATE.md × [🔍] [🔍] [🔍] [←] [→] [⋮]
.github > PULL_REQUEST_TEMPLATE.md > [abc] ## Summary of changes
1 <!-- Wow, more E2E tests, sweet! -->
2 ## Summary of changes
3
4 -
5 - [ ] are the tests readable?
6
7 ## Any linked Jira or GitHub issues?
8
9 -
10
11 ## Can the reviewer merge the pull request after approval?
12
13 - [ ] yes, the reviewer can merge the pull request after approval and the new or updated tests passed on CI
14
15 ## Can any MABL / other automation tests be retired?
16
17 - list the title or link to the test to be retired after merging this pull request
18
```

Pull request template

- **Keep the tests readable**
 - **custom commands, utilities, plugins**

```

  v support
    TS assertions.d.ts
    JS assertions.js
    TS commands.d.ts
    JS commands.js
    TS delay.d.ts
    JS delay.js
    TS get-by.d.ts
    JS get-by.js
    TS index.d.ts
    JS index.js
    JS utils-helpcenter.js
    JS utils.js
    JS zendesk.js

```


- Keep the tests readable
 - custom commands, utilities, plugins
- Do as much as possible via API calls
 - login, add an address, add a credit card, create a listing, etc
- **Cache created data**
 - **users, listing**



<https://slides.com/bahmutov/flexible-cypress-data>


- Keep the tests readable
 - custom commands, utilities, plugins
- Do as much as possible via API calls
 - login, add an address, add a credit card, create a listing, etc
- Cache created data
 - users, listing
- **Keep a test shorter than 3 minutes**
 - **Keep each spec shorter than 3 minutes**
 - **Use data-driven testing**


```
1 const searches = ['Wearable', 'Running shoes',  
  'Dolls']  
2  
3 it.each(searches)(  
4   `Filters results by status for search: %s`,  
5   (searchKeyword) => {  
6     const url = formSearchUrl({ searchKeyword })  
7     cy.visitProtectedPage(url)  
8     ...  
9   })
```

<https://github.com/bahmutov/cypress-each>

Test suite size

FILTER BY

 main ▾

 Tag: Any ▾

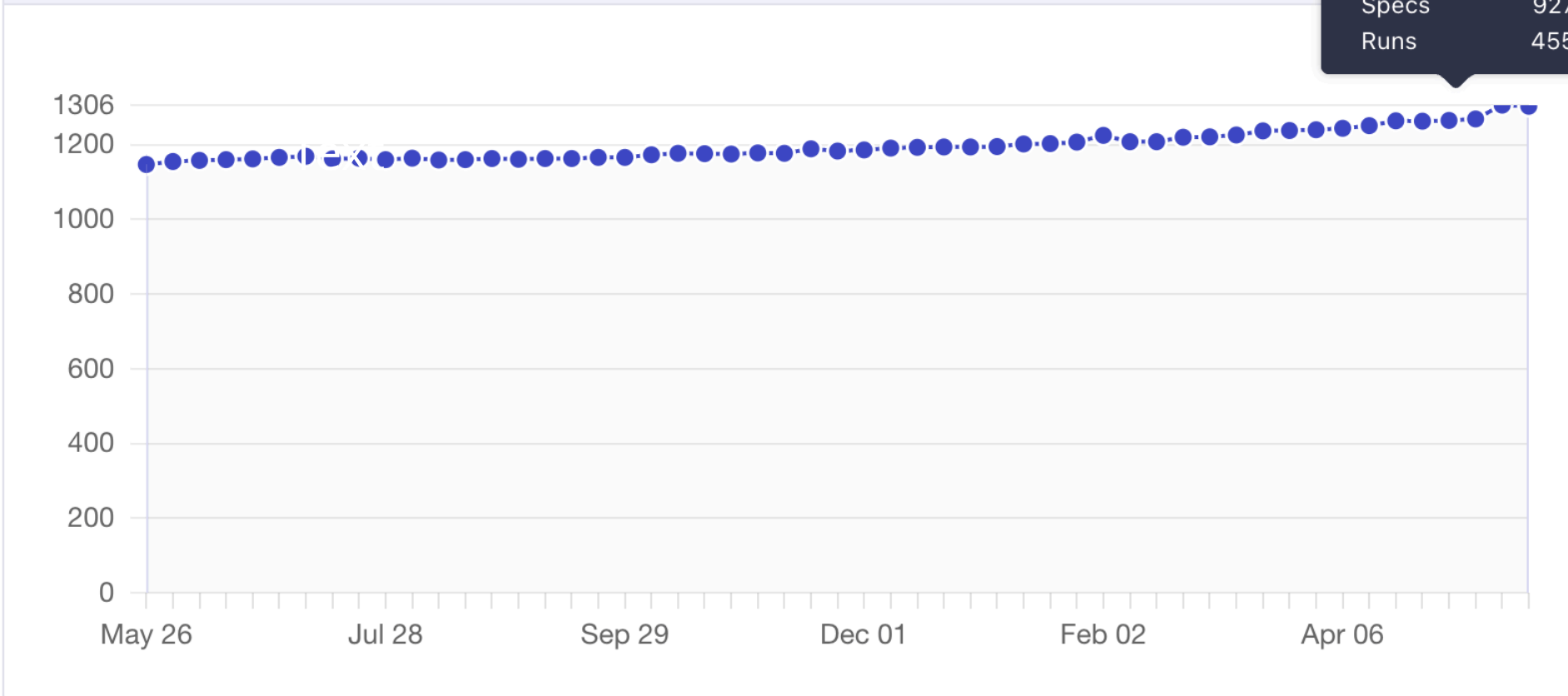
 Last 12 Months ▾

 Weekly ▾

 Reset Filters



Test suite size over time





Why E2E Tests
when we have
unit tests?



It Is
A Question
Of Scale

1303 tests * 1 minute/test

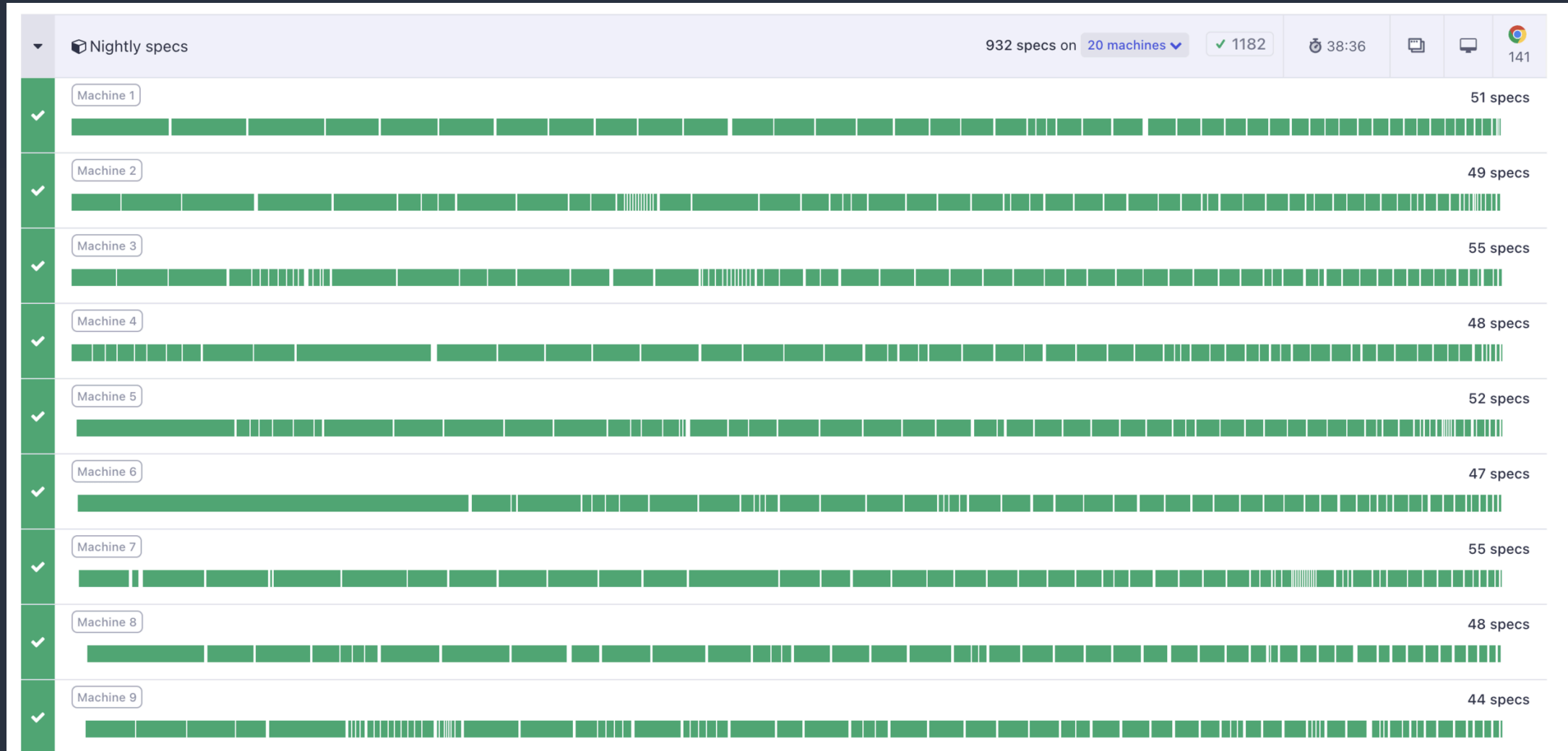
≈

22 hours to run all the tests

1. How to run all the tests?
2. How to run the tests for PRs?



Parallelize all the things



E2E tests finish in 39 minutes using 20 CI machines

```
1 # .github/workflows/nightly.yml
2 nightly-tests:
3   needs: [prepare]
4   container: cypress/browsers:node-22.20.0-chrome-141.0.7390.107-1-ff-
144.0-edge-141.0.3537.85-1
5   strategy:
6     fail-fast: false
7     matrix: ${{fromJSON(needs.prepare.outputs.matrix)}}
8   steps:
9     - name: Cypress nightly tests 🛠
10      uses: cypress-io/github-action@6.6.1
11      with:
12        record: true
13        parallel: true
```

Use the Cypress GitHub Action



Dev Environment



🔥 Dev Environment 🔥
🔥 🔥 🔥 🔥 🔥

(cannot run 1000 tests in parallel)

An aerial photograph of a two-lane road. The top lane is clear, with a single white van driving away from the viewer. The bottom lane is completely blocked by a long, single-file line of various colorful trucks and trailers, including red, blue, orange, and white ones. The road is flanked by green grass and some dry, brown bushes.

We have to prioritize some
spec files

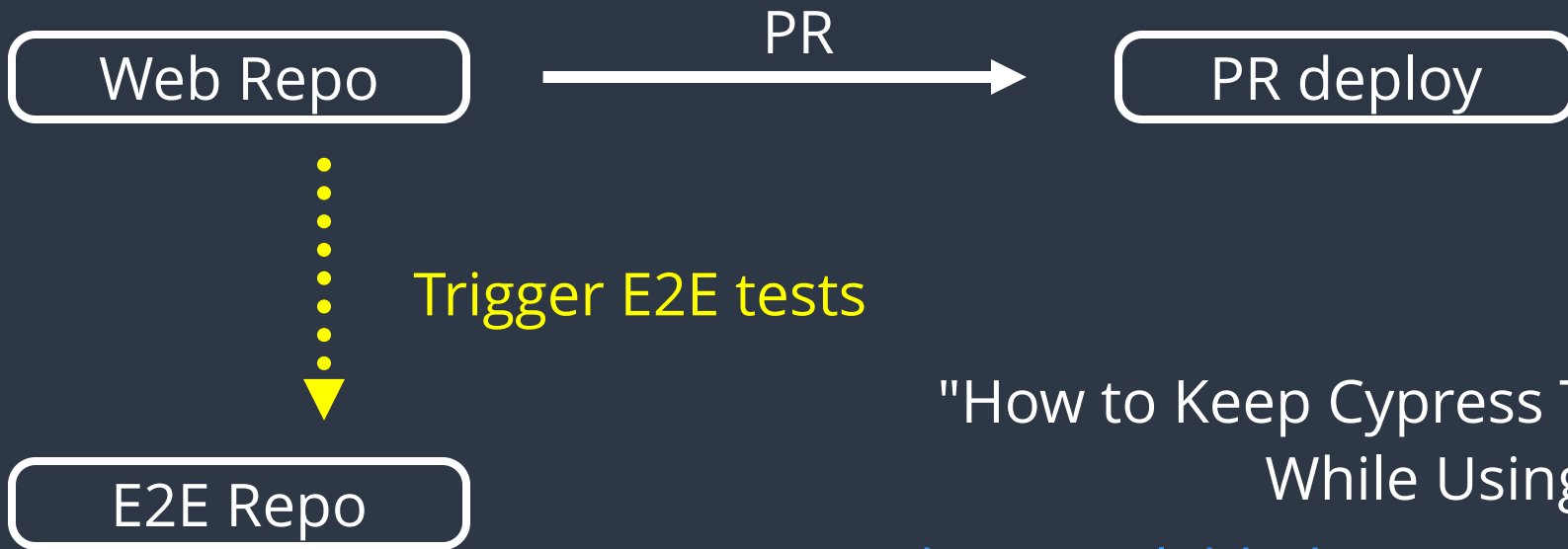
A dark blue ceramic coffee cup filled with coffee, sitting on a matching saucer. The cup and saucer are placed on a rustic wooden table. The lighting is warm and focused on the cup, creating soft shadows.

Work locally
on a single
feature spec

A photograph of a dark grey ceramic coffee cup filled with coffee, sitting on a matching saucer. The cup and saucer are placed on a rustic, dark brown wooden table. The lighting is warm and directional, coming from the upper left, creating soft shadows and highlighting the textures of the wood and the ceramic. The background is blurred, focusing attention on the coffee cup.

Push code
to run E2E
tests on CI

Pull Request Flow



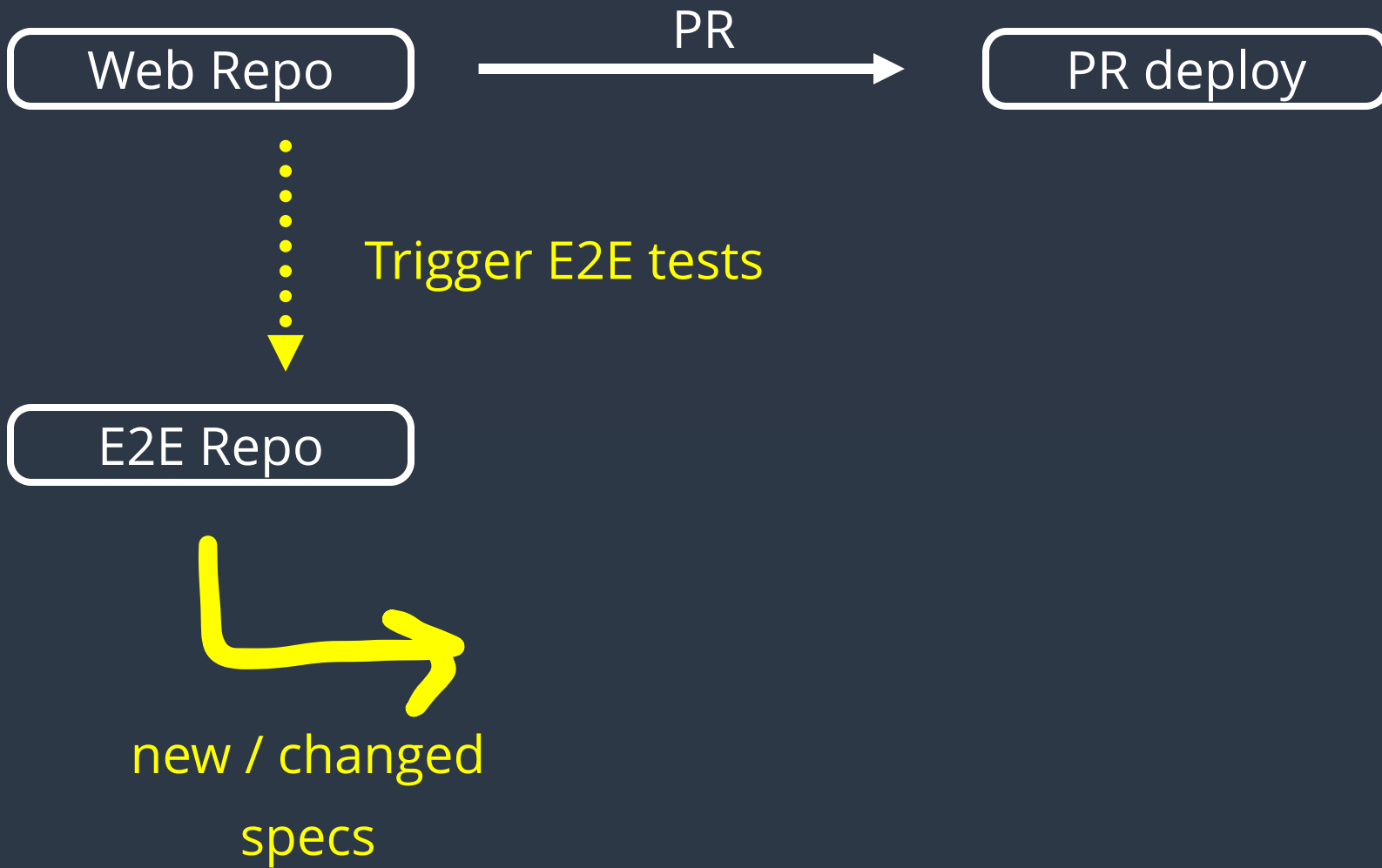
"How to Keep Cypress Tests in Another Repo While Using CircleCI"

<https://glebbahmutov.com/blog/how-to-keep-cypress-tests-in-another-repo-with-circleci>

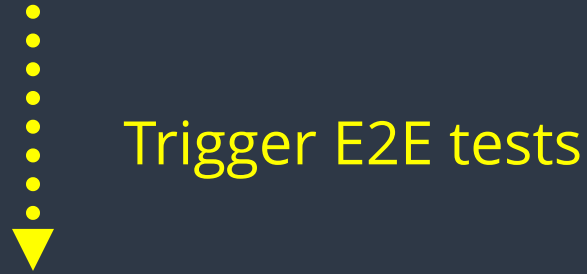
"How to Keep Cypress Tests in Another Repo While Using GitHub Actions"

<https://glebbahmutov.com/blog/how-to-keep-cypress-tests-in-another-repo/>

Pull Request Flow



Pull Request Flow



CI machines



// PR preview environments are isolated (GOOD), but not very powerful even compared to the DEV environment (ughh)

Find the changed specs and run them first

```
1 # https://github.com/bahmutov/find-cypress-specs
2 specs=$(npx find-cypress-specs --branch main --parent)
3 n=$(npx find-cypress-specs --branch main --parent --count)
4
5 if [ ${n} -lt 1 ]; then
6   echo "No Cypress specs changed, exiting..."
7   exit 0
8 fi
9
10 npx cypress run --record --parallel --spec ${specs}
```



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



If changed specs
pass, run all or
some E2E tests

Bonus 100

- Pick / run tests based on frontend code changes
 - <https://glebbahmutov.com/blog/using-test-ids-to-pick-specs-to-run/>
- Pick / run tests based on visited pages
 - <https://glebbahmutov.com/blog/run-cypress-tests-for-the-given-url/>
- Pick / run tests based on API calls
 - <https://glebbahmutov.com/blog/pick-tests-by-network-calls/>

Test 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://github.com/bahmutov/cy-grep>

<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://github.com/bahmutov/cy-grep>

<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 @balance     4
6 @careers     2
7 @helpcenter  69
8 @local       19
9 @login       11
10 @messaging   8
11 @mobile      77
12 @moble       1
13 @offer       6
14 @payment     34
15 @profile     55
16 @regression  72
17 @sanity      24
18 @search      43
19 @sell        76
20 @shipping    24
21 @signup      12
22 @w9          10

```

 Learn more:

<https://github.com/bahmutov/cy-grep>

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

<https://github.com/bahmutov/find-cypr>

@profile

@sell

@login

@shipping

@payment

@...

■ ■ ■ ■ ■ ■ ■ @sanity

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

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

- Run all tests 3x a day
- Run each test tag in its own workflow once a day
- Run high priority test tags like @payment and @shipping 2-3x a day



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

Trigger tests from GitHub UI

<https://glebbahmutov.com/blog/cypress-grep-filters>

<https://glebbahmutov.com/blog/pick-tests-using-pull-request/>

Use workflow from

Branch: main ▾

Run the tests tagged "@sanity", small set

Run all the tests

Other test tags to run, comma-separated. Includes @profile, @signup, @search, @login, @mobile, @offer, @payment, @shipping, @sell, @local, @helpcenter, @balance, @w9, @careers, @messaging

The base URL to run tests against

The HelpCenter custom URL if any

Custom API URL to pass as X-Mercari-API-Endpoint

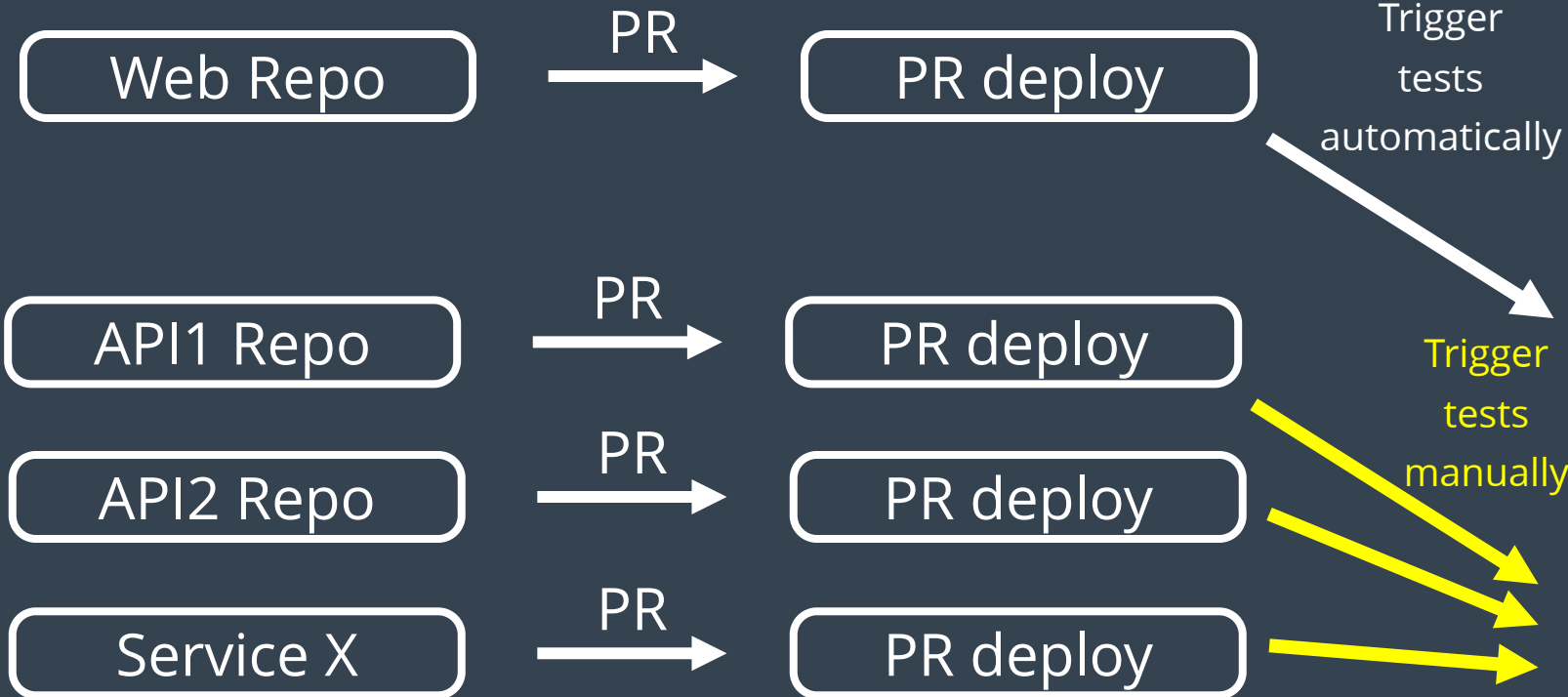
Custom GraphQL URL PR number to use during testing

Number of machines to use

Cypress Dashboard run tag

Send test results to TestRail

YOU can run the tests



Use workflow from

Branch: main ▾

Run the tests tagged "@sanity", small set

Run all the tests

Other test tags to run, comma-separated. Includes @profile, @signup, @search, @login, @mobile, @offer, @payment, @shipping, @sell, @local, @helpcenter, @balance, @w9, @careers, @messaging

The base URL to run tests against

The HelpCenter custom URL if any

Custom API URL to pass as X-Mercari-API-Endpoint

Custom GraphQL URL PR number to use during testing

Number of machines to use

Cypress Dashboard run tag

Send test results to TestRail



jjhelguero commented 5 days ago

Author



/cypress



jjhelguero marked this pull request as ready for review 5 days ago



github-actions bot commented 5 days ago



 Running Cypress

- Job: <https://github.com/org/repo/actions/runs/3070711466>
- Command: cypress
- Tags:

Custom /cypress command in the PR comment for all repos

```
/cypress tags=@shipping,@profile
```



Test Tags AI

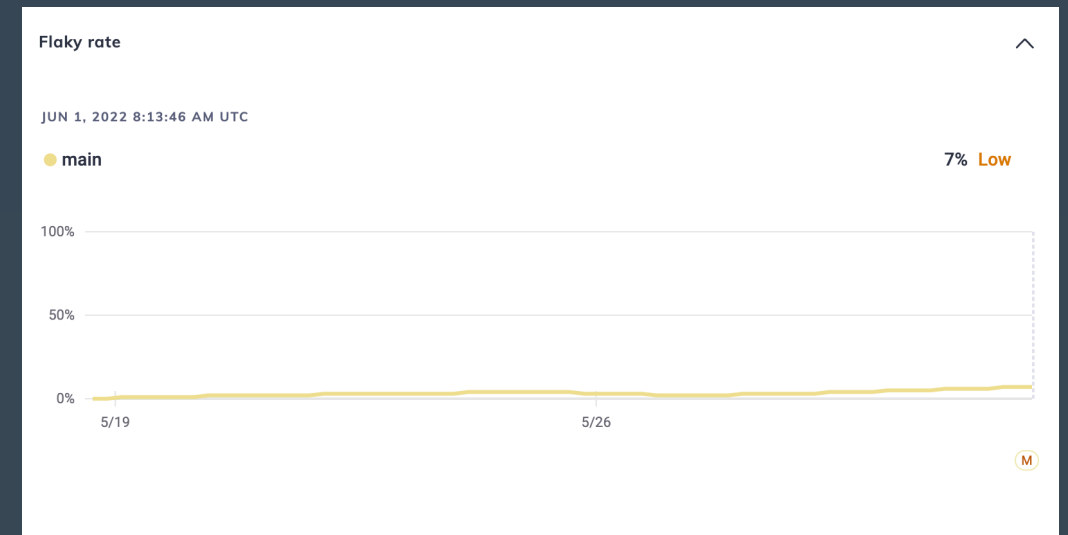
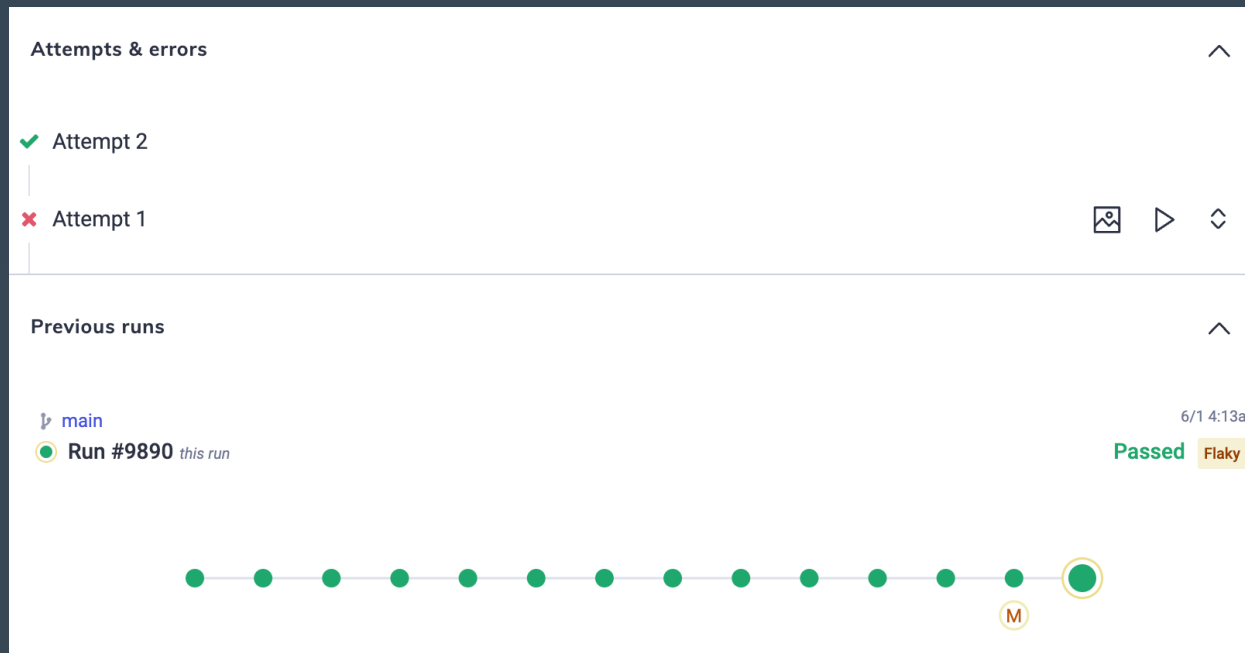
// AI, what is the best test tag(s) for this pull request?

Flake Is Bad

Fighting flaky tests

<https://on.cypress.io/flaky-test-management>

- Enable test retries, look at the Cypress Dashboard



Fighting flaky tests

<https://on.cypress.io/flaky-test-management>

- Enable test retries, look at the Cypress Dashboard
- Increase command timeouts where necessary
- Command - assertion pattern

Fighting flaky tests

<https://on.cypress.io/flaky-test-management>

- spy on GraphQL calls 🎉

"Directly Spying on GraphQL Calls Made By The Application"

<https://www.youtube.com/watch?v=XadOqS0YNJE>

spy on GraphQL calls 🎉

The screenshot shows the Cypress test runner interface. On the left, the command log displays a sequence of actions: assertions for visibility and text, and several 'get' commands targeting elements with test IDs like 'ShippingFee', 'Tax', and 'CheckoutButton'. The final command is a '-click' action on the 'CheckoutButton' at line 397, which is highlighted in red. Below the command log, a network log shows multiple 'fetch' and 'xhr' requests, all with a status of 'req modified no alias'. On the right, the browser view shows a 'MERCARI Secure checkout' page with sections for 'Shipping address', 'Payment', and 'Delivery method'. An 'Orders summary' box is visible on the right side of the page, containing item details and a 'Checkout' button. An orange arrow points from the '-click' command in the log to the 'Checkout' button in the browser view.

```
Attempt 1
CypressError
2022-06-01T08:20:00.099Z
Timed out retrying after 20050ms: cy.click() failed because this element is detached from the DOM.
<button type="submit" data-testid="CheckoutButton" class="Button_BaseButton-sc-11w253b-0 eNGBTX">...
</button>
Cypress requires elements be attached in the DOM to interact with them.
The previous command that ran was:
> cy.get()
This DOM element likely became detached somewhere between the previous and current command.
Common situations why this happens:
• Your JS framework re-rendered asynchronously
• Your app code reacted to an event firing and removed the element
You typically need to re-query for the element or add 'guards' which delay Cypress from running new commands.
https://on.cypress.io/element-has-detached-from-dom
cypress/integration/data-sessions.js at line 179
177 |         cy.byTestId('Tax').should('be.visible').and('not.have.text', '--')
178 |
> 179 |         cy.byTestId('CheckoutButton').click()
```

spy on GraphQL calls 🎉

```
154     cy.interceptGraphQL([
155       'transactionInitQuery',
156       'transactionBraintreeQuery',
157       'salesTaxQuery',
158 +     'DeliveryAddresses',
159     ])
160     getBuyNowButton().click()
161     cy.location('pathname')
162 ) {
176       .and('not.have.text', '--')
177       // Wait for tax to be fetched, page should be stable after that.
178       cy.byTestId('Tax').should('be.visible').and('not.have.text', '--')
179 +     cy.wait('@DeliveryAddresses')
180
181     cy.byTestId('CheckoutButton').click()
182     cy.byTestId('Modal', 'Nearly there').should('be.visible').delay()
```

"Set GraphQL Operation Name As Custom Header And Use It In cy.intercept"
<https://www.youtube.com/watch?v=AcU5mkedchM> deserves a lot more ❤️

Prevent Flaky Tests:

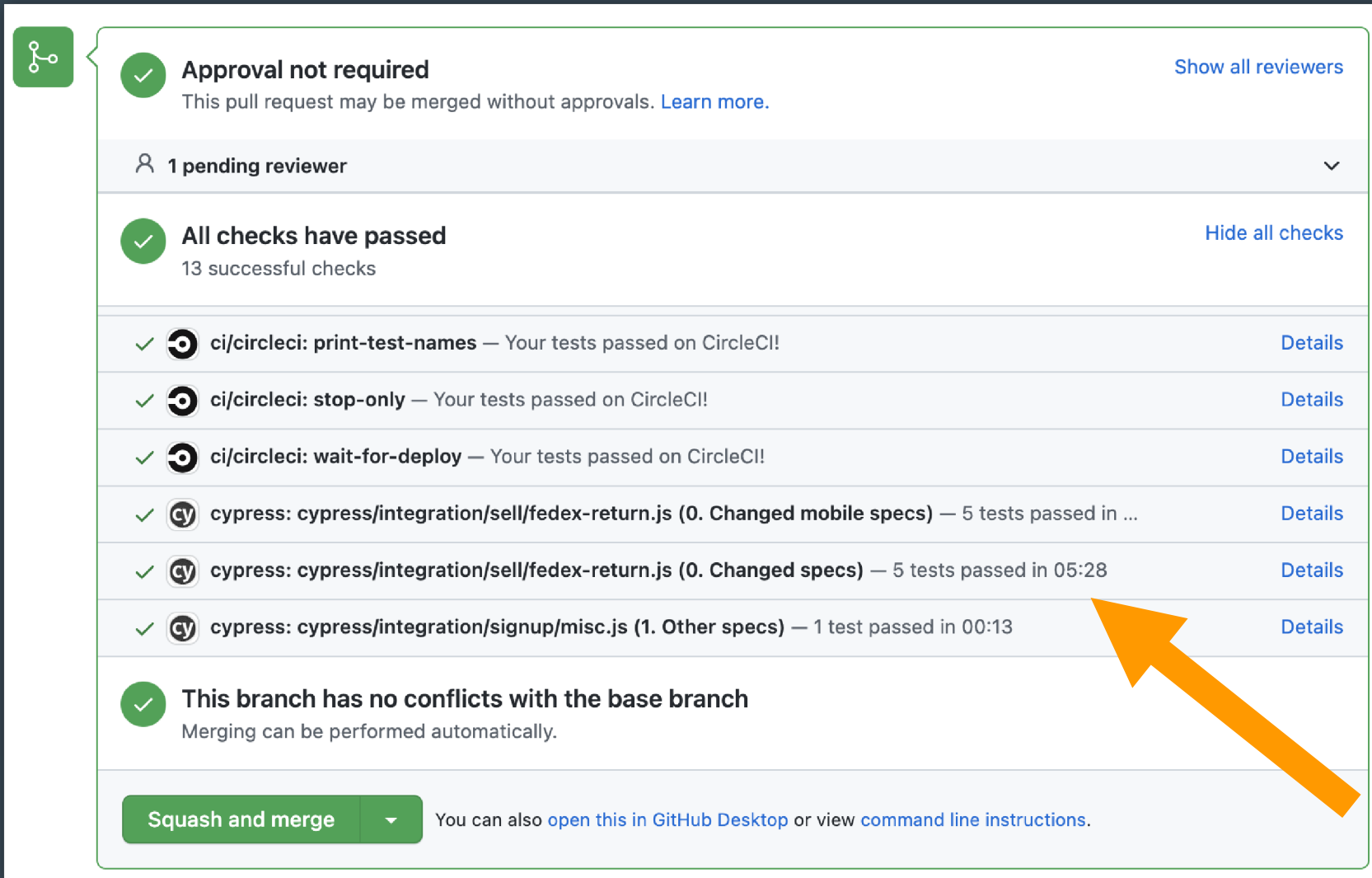
Burn New Tests






```
CYPRESS_burn=5 npx cypress run ...
```

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







Burn Changed / New Tests




 **Approval not required** [Show all reviewers](#)
This pull request may be merged without approvals. [Learn more.](#)

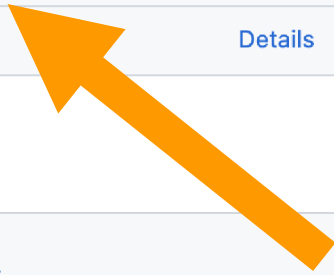
 **1 pending reviewer** 

All checks have passed [Hide all checks](#)
13 successful checks



-  **ci/circleci: print-test-names** — Your tests passed on CircleCI! [Details](#)
-  **ci/circleci: stop-only** — Your tests passed on CircleCI! [Details](#)
-  **ci/circleci: wait-for-deploy** — Your tests passed on CircleCI! [Details](#)
-  **cypress: cypress/integration/sell/fedex-return.js (0. Changed mobile specs)** — 5 tests passed in ... [Details](#)
-  **cypress: cypress/integration/sell/fedex-return.js (0. Changed specs)** — 5 tests passed in 05:28 [Details](#)
-  **cypress: cypress/integration/signup/misc.js (1. Other specs)** — 1 test passed in 00:13 [Details](#)


This branch has no conflicts with the base branch
Merging can be performed automatically.


Squash and merge  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).










Burn Changed / New Tests

  **Approval not required** Show all reviewers
This pull request may be merged without approvals. [Learn more.](#)

 1 pending reviewer



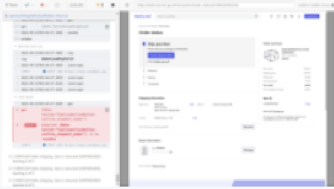
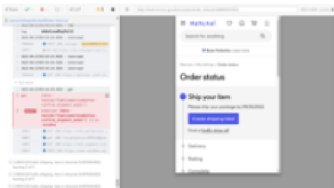
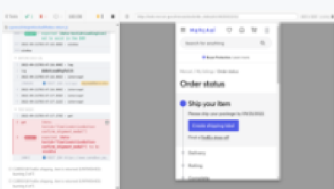
 **All checks have passed**
13 successful checks

-  **ci/circleci: print-test-names** — Your tests passed on CircleCI!
-  **ci/circleci: stop-only** — Your tests passed on CircleCI!
-  **ci/circleci: wait-for-deploy** — Your tests passed on CircleCI!
-  **cypress: cypress/integration/sell/fedex-return.js (0. Change**
-  **cypress: cypress/integration/sell/fedex-return.js (0. Change**
-  **cypress: cypress/integration/signup/misc.js (1. Other spec**

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

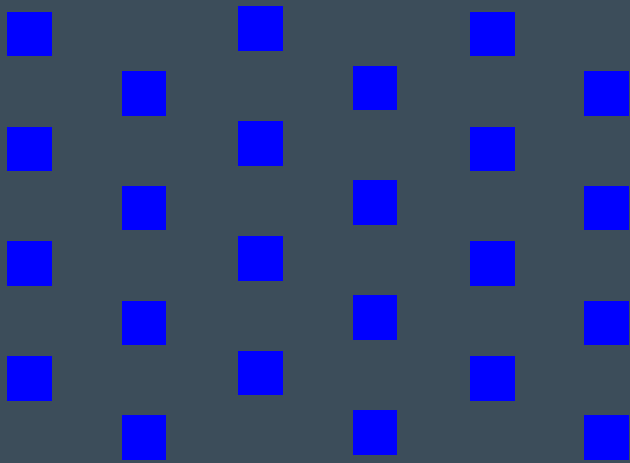
Squash and merge You can also [open this in GitHub Desktop](#)

Flakiness

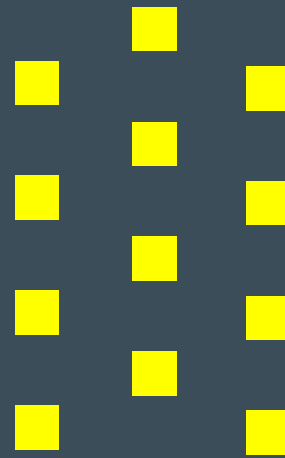
	cypress/integration/sell/fedex-return.js 	3 Flaky
1	SELLER > test name: burning 1 of 5	
2	SELLER > test name (mobile): burning 1 of 5	
3	SELLER > test name (mobile): burning 2 of 5	

Where are we going...

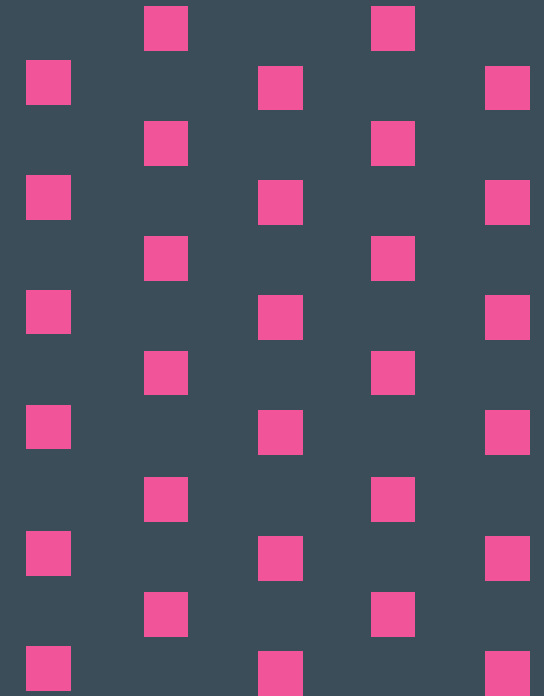
Automated E2E Tests



Manual E2E Tests



API / unit tests



← cheaper!
faster!
never tired!

What we did:

- Cypress demo sessions
- **Writing example tests for any discovered errors**
- Writing and sharing a LOT of blog posts / examples

Test replays via Cypress Cloud

Report an issue

```

GUEST > Tapping Notification
C12190494 Sell now leads to login page

144 -assert expected [data-testid="AlertDialog"] not to exist
    in the DOM
145 2026-06-01T14:11:09.441Z - byTestId
146 2026-06-01T14:11:09.442Z - contains
147 -contains [data-testid="HeaderTitle"], List an item
148 2026-06-01T14:11:09.445Z - byTestId
149 2026-06-01T14:11:09.446Z - get
150 get [data-testid="EarnMoreWithSellerTierModal"]
151 assert expected undefined to exist
152 2026-06-01T14:11:09.449Z - byTestId
153 2026-06-01T14:11:09.450Z - contains
154 -contains [data-testid="HeaderTitle"], List an item
155 -assert expected <h1.H1-sc-1a7mju.FormLayout__HeaderTitle-
    sc-30460404-2.hGtLxB.cQYKaM> to be visible
156 2026-06-01T14:11:09.454Z - byTestId
157 2026-06-01T14:11:09.454Z - get
158 get [data-testid="LoadingIcon"]
159 -assert expected [data-testid="LoadingIcon"] not to exist
    in the DOM
160 2026-06-01T14:11:09.456Z - byTestId
161 2026-06-01T14:11:09.457Z - get
162 get [data-testid="PhotoUploadBox"]
163 -assert expected <div.Flex-sc-638440ba-0.bFTpDX> to be
    visible

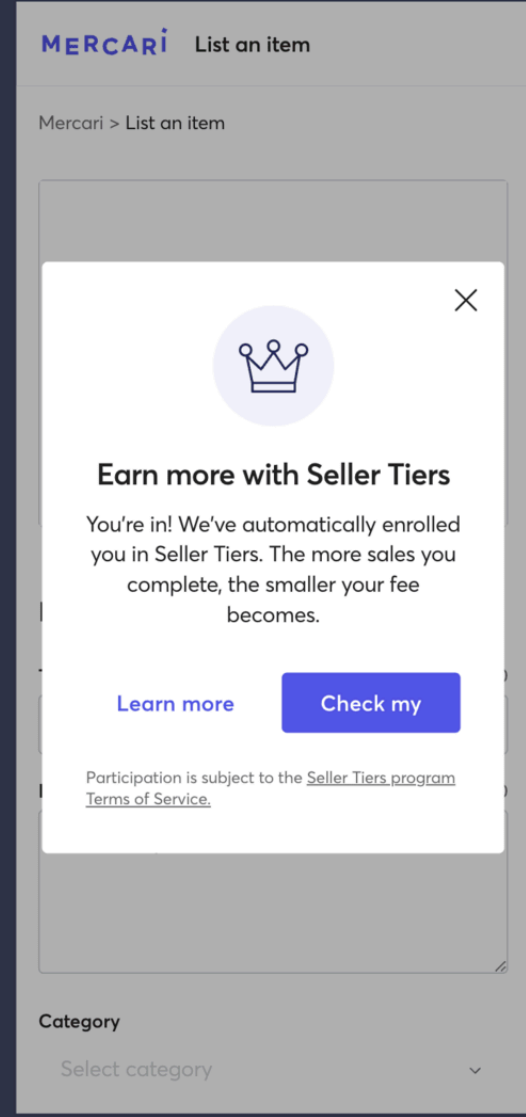
(image) GET 200 https://u-web-
assets.dev5s.com/assets/icons/sellerTie
rs/icn-crown.svg?width=64&quality=75 req modified no alias
(fetch) POST 204 https://www.google-
analytics.com/g/collect?v=2&tid=G-
W39NDQ7PVW>m=45je65r2v875184990za200zb7 req modified no alias

```

https://web-pr10412.mercari.guru/sell/

375x812

Network Console Inspect



All	Errors	Fetch/XHR	Filter
200	POST	api?operationName=TrackEvent&operationNam...	>
200	GET	m=credential_button_library	>
200	GET	m=credential_button_library	>
200	POST	api?operationName=sellUserInfoQuery	>
200	POST	api?operationName=masterQuery	>
200	POST	bcn?k=6LfXaxEaAAAAAlum0BWsU-0xRYO_5wo...	>
200	GET	icn-crown.svg?width=64&quality=75	>
201	POST	session-data	>
204	POST	collect?v=2&tid=G-W39NDQ7PVW>m=45je6...	>
204	POST	collect?v=2&tid=G-W39NDQ7PVW>m=45je6...	>
200	GET	arrow.svg	>
200	GET	api?operationName=sellQuery&variables=%7B%...	>
200	POST	api?operationName=userLastListedItemsQuery	>
200	POST	api?operationName=TrackEvent	>
200	POST	api?operationName=TrackEvent	>
200	GET	api?operationName=smartSalesFeeQuery&variab...	>
204	POST	collect?v=2&tid=G-W39NDQ7PVW>m=45je6...	>
201	POST	https://sdk.iad-01.braze.com/api/v3/data/	>
200	GET	api?operationName=sellerAcceptedOfferConvers...	>
--	POST	session-data	>
--	POST	collect?v=2&tid=G-W39NDQ7PVW>m=45je65r...	>
--	POST	collect?v=2&tid=G-W39NDQ7PVW>m=45je65r...	>
--	POST	session-data	>

What we have achieved:

- Web team is writing / updating tests
- **AI is writing / updating tests**
- Everyone looks at web test runs and results

“ We now catch bugs the same day or even before they are merged and deployed

recent Mercari US QA team meeting

The End



<https://slides.com/bahmutov/1000-e2e>

gleb.dev

