

Write E2E Test for Mattermost Plugin

With Modern E2E Testing Framework Cypress

Press Space for next page →



はじめに

- 本日の資料はSlidevで作成しています
 - Vue コアチームの Anthony Fu が開発
 - Markdown to Slide
 - カスタムVueコンポーネントを使える
 - SPA としてビルド可能

Q: 本日のスライドのどこにカスタムVueコンポーネントが使われているでしょう？



Slidev

Why E2E Testing?

Why E2E Testing?

BACKGROUND

- Mattermost でプラグインを書いている
- Mattermost プラグインは、Mattermost 上でのユーザーのアクションに応じて実行される処理を記述できる
 - **MessageWillBePosted**: 投稿されたメッセージが DB に保存される前に実行される処理
 - **UserHasJoinedChannel**: ユーザーがチャネルに参加した時に実行される処理
 - => ユーザーが多いインスタンスでの負荷が気になる
- Mattermost Webapp 側のプラグインでは、処理をミスると画面全体がホワイトアウトする
 - => ユーザーへの影響が大きいため、導入を躊躇する
- ユニットテストでは Mattermost と組み合わせたときの動作を確認できない

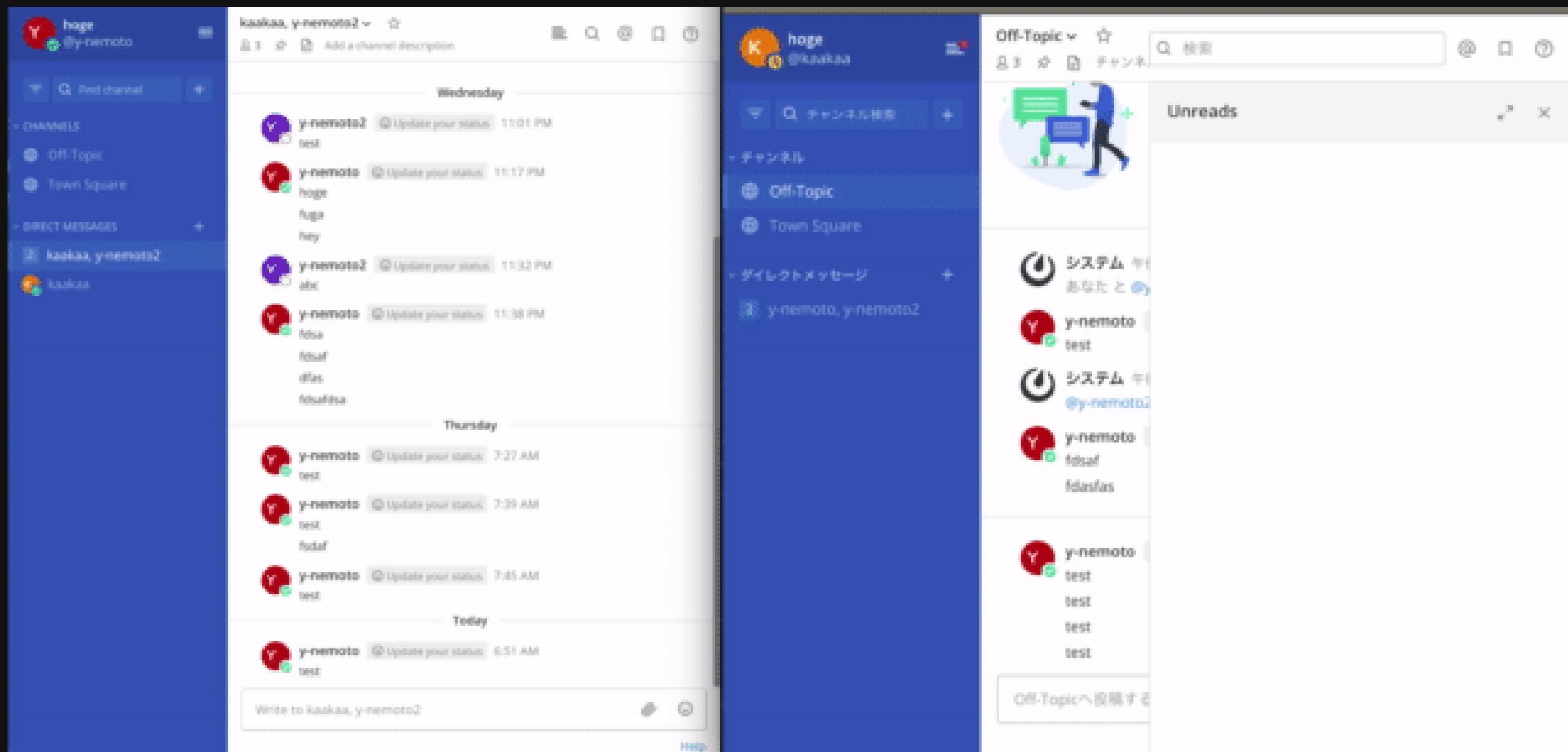
E2E テストを書きたい

- Mattermost と組み合わせた場合の自動テストを書いておきたい
 - リグレッションを避けるため
 - パフォーマンスへの影響を計測するため

Why E2E Testing?

TARGET PLUGIN

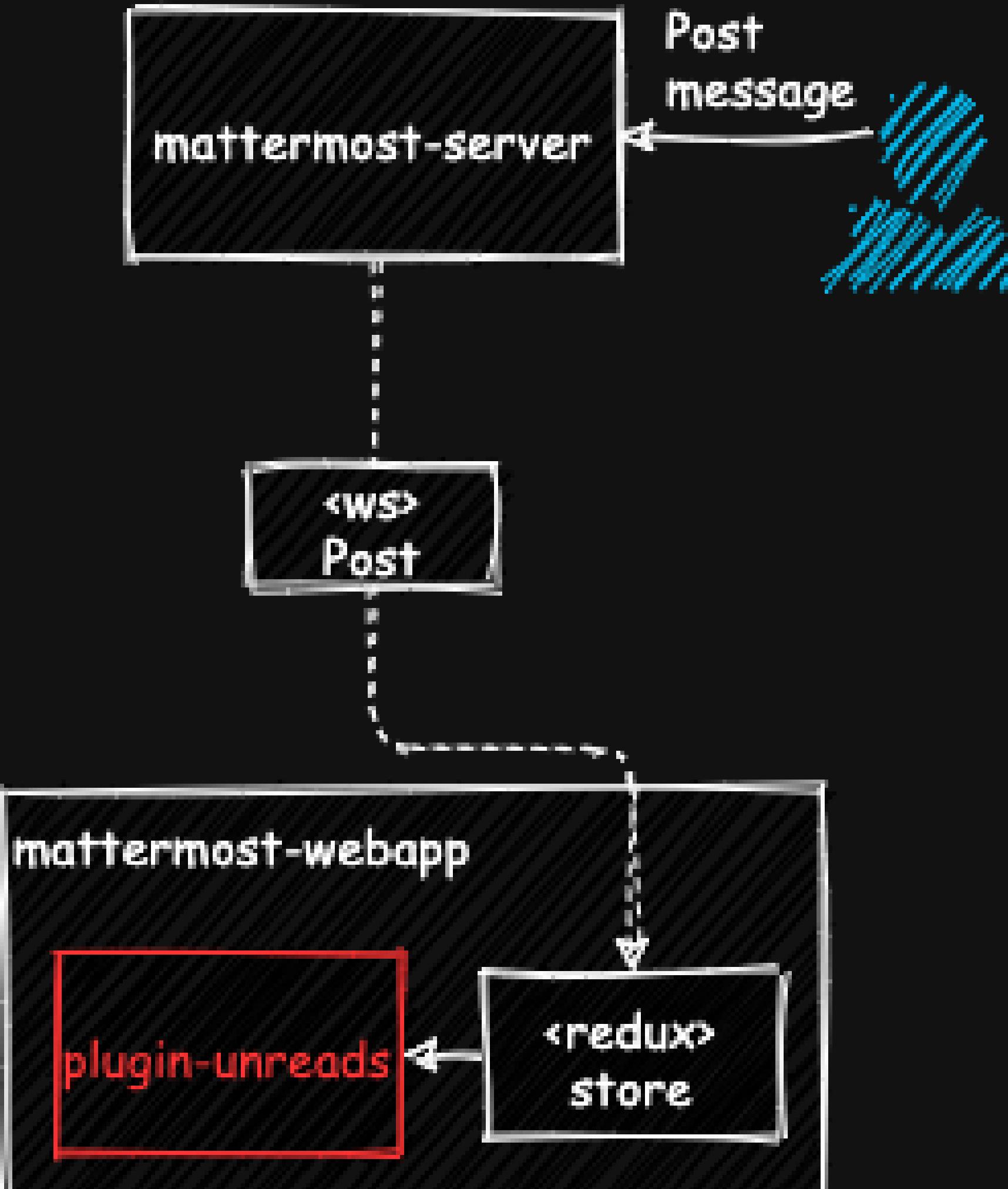
- kaakaa/mattermost-plugin-unreads
 - Slackで言う `All unread` の機能



Why E2E Testing?

TARGET PLUGIN

- kaakaa/mattermost-plugin-unreads
 - Slackで言う `All unread` の機能
 - mattermost-reduxを使ってStoreのPostにアクセス
 - Mattermost Plugin WebApp API の registerRightHandSidebarComponentでレンダリング
- JestでSnapshot Testはできるが、Mattermostと組み合わせてテストしたい



Why E2E Testing?

CYPRESS IN MATTERMOT PROJECT

- MattermostではCypressでE2Eテストを書いている
 - たまに E2E Test Automation Hackfest と題して Cypress のテストのコントリビューションを集めている
- いくつかのプラグインでは、Cypress を使った E2E テストを書いている
 - mattermost/mattermost-incident-collaboration
 - mattermost/mattermost-plugin-apps

E2E Test Automation Hackfest 2021 🚀 #17555

(! Open) prapti opened this issue 11 days ago · 0 comments

prapti commented 11 days ago · edited

Cypress Test Automation Hackfest 🚀

Win an exclusive Mattermost swag bag by making five contributions to the Cypress Test Automation campaign during the month of May, 2021.
Plus top-3 contributors receive a special award!

The Mattermost team is working on automating feature testing of the webapp to

1. Decrease time to ship a bug fix release
2. Raise the quality of the software by reducing number of manual tests

We're excited to work with our community to significantly increase automated test coverage! In addition to making a huge impact on the project, you'll get experience working with Cypress, one of the most popular open source test automation frameworks.
If it's your first time contributing to Mattermost, you'll also receive a [custom-printed Mattermost Mug!](#)

Cypress で Mattermost プラグインのテストはどう書く？

What is Cypress?



Cypress

- Web アプリ向け? の E2E Testing Framework
 - Selenium的なモノ?
- 最近のデファクトスタンダードな雰囲気

The web has evolved.
Finally, testing has too.

Fast, easy and reliable testing for anything that runs in a browser.



\$ npm install cypress

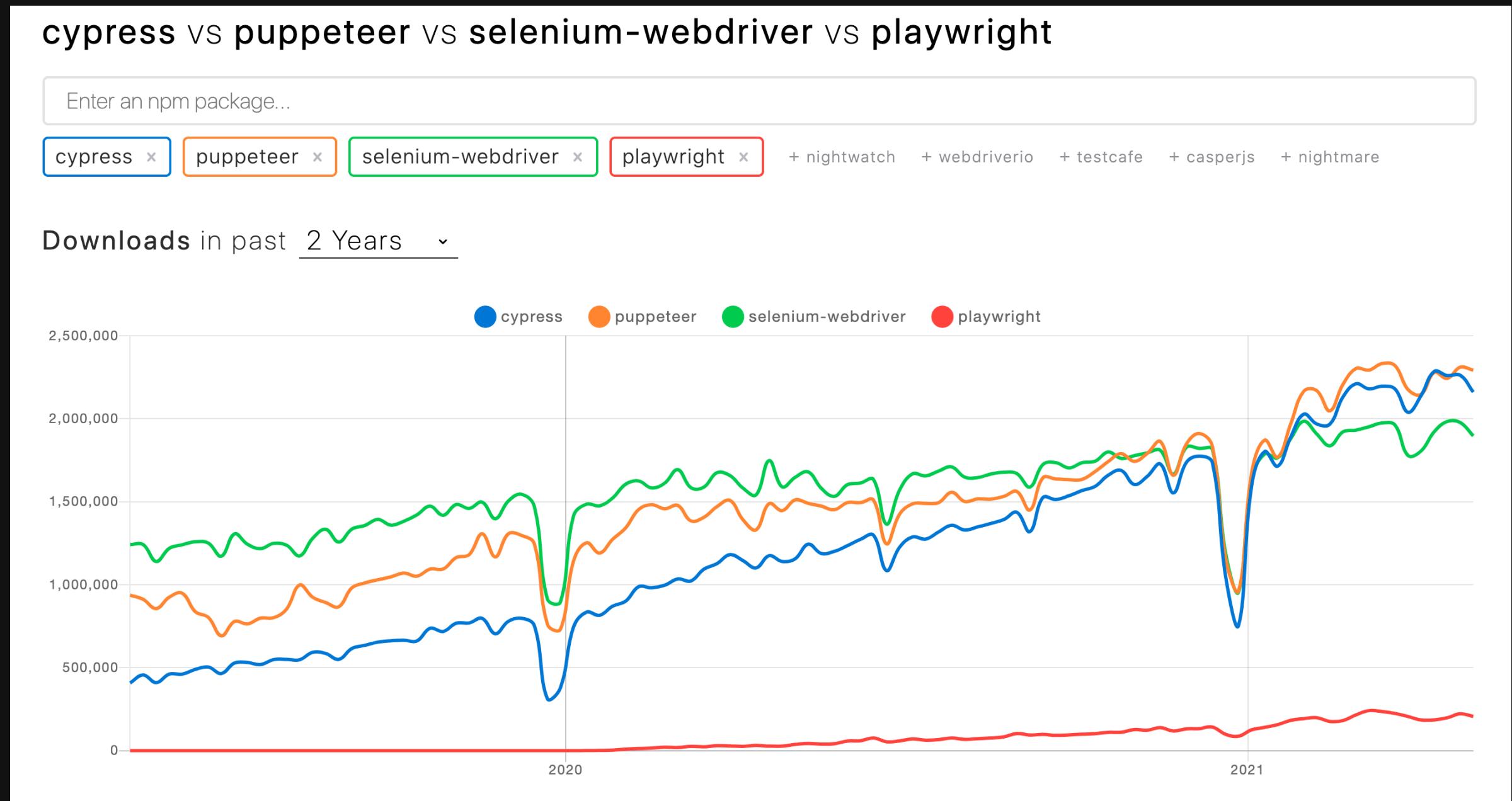
or [Download now](#) ?

Install Cypress for Mac, Linux, or Windows, then [get started](#).

A screenshot of a web browser window. On the left, the Cypress Test Runner interface shows a test titled "looks good on mobile" with several test steps listed. The main content area displays a social media feed from a user named Dave Gamache. His profile picture is a black and white portrait of a man with glasses. His bio includes a desire to be taller and a mention of being a baller. He has 12M friends and 1 enemy. Below his bio are links to his About section and a list of accomplishments: Went to Oh, Canada; Became friends with Obama; Worked at Github; Lives in San Francisco, CA; and From Seattle, WA. To the right of his profile is a large image of a person standing on a rocky cliff overlooking a vast, misty landscape. Further down the page are sections for Sponsored posts (a yellow building in Iceland) and Likes (Jacob Thornton's profile).

Cypress

COMPARE - NPM TRENDS



- 2021年から`Cypress`, `Puppeteer` > `Selenium`
- `Playwright`は`Puppeteer`の牙城を崩せておらず

Cypress

COMPARE - GITHUB

COMPARE - GITHUB

selenium

Repository	Owner	Stars	Forks	Open issues	Age	Last commit	License	Language
cypress-io/cypress	cy	30,476	1,776	1,389	6 years ago	a day ago	MIT	JavaScript
puppeteer/puppeteer	Puppeteer	70,559	7,403	1,604	4 years ago	a day ago	Apache-2.0	TypeScript
microsoft/playwright	Microsoft	23,860	963	226	a year ago	a day ago	Apache-2.0	TypeScript
SeleniumHQ/selenium	SeleniumHQ	20,765	6,309	182	8 years ago	a day ago	Apache-2.0	Java

How to run Cypress?

Cypress

SETUP

```
# Install
$ npm i cypress
```

```
# Open DesktopApp
$ node_modules/.bin/cypress open
```

- **cypress open** でテスト管理用デスクトップアプリが開く
- spec ファイルをクリックするとテスト実行
- spec ファイル追加・修正を自動で反映
- 右上メニューから起動ブラウザを選択可能

tesmp

Tests Runs Settings

We've created some sample tests around key Cypress concepts. Run the first one or create your own test file.

How to write tests

Press Cmd + F to search...

+ New Spec File

INTEGRATION TESTS COLLAPSE ALL | EXPAND ALL

examples

actions.spec.js
aliasing.spec.js
assertions.spec.js
connectors.spec.js
cookies.spec.js
cypress_api.spec.js
files.spec.js
local_storage.spec.js
location.spec.js
misc.spec.js
navigation.spec.js
network_requests.spec.js
querying.spec.js
spies_stubs_clocks.spec.js
traversal.spec.js
utilities.spec.js
viewport.spec.js
waiting.spec.js
window.spec.js

Run 19 integration specs

Cypress

TEST RUNNER

- specファイルをクリックすると、テスト用のブラウザが起動しテストが実行される
- `cy` オブジェクト

```
//> <reference types="cypress" />

context('Google', () => {
  beforeEach(() => {
    cy.visit('https://www.google.com/');
  });

  it('github finds https://github.com/', () => {
    cy.get('input.gLFyf.gsfi')
      .type('github{enter}');

    cy.get('a[href="https://github.com/"]')
      .should($anchor) => {
        expect($anchor).to.have.length(1);
      });
  });
});
```

The screenshot shows the Cypress Test Runner interface. On the left, the test code is displayed:

```
cypress/integration/examples/google.spec.js
```

The test context is 'Google'. It includes a `beforeEach` block that visits `'https://www.google.com/'`, which triggers several XHR requests for autocomplete suggestions. The main `it` block types 'github' into the search bar, triggering more XHR requests for search results. Finally, the test asserts that there is one anchor element with a href to GitHub.

On the right, the actual Google search results page is shown, matching the search term 'github'.

Cypress

TEST CODE

```
/// <reference types="cypress" />

context('Google', () => {
  beforeEach(() => {
    cy.visit('https://www.google.com/');
  });

  it('github finds https://github.com/', () => {
    cy.get('input.gLFyf.gsfi')
      .type('github{enter}');

    cy.get('a[href="https://github.com/"]')
      .should(($anchor) => {
        expect($anchor).to.have.length(1);
      });
  });
});
```

```
cypress/integration/examples/google.spec.js
```

```
Google
```

```
✓ github finds https://github.com/
```

```
▼ BEFORE EACH
```

```
visit https://www.google.com/
```

```
▼ TEST BODY
```

```
1 get input.gLFyf.gsfi
```

```
2 -type github{enter}
```

```
(xhr) ● GET (canceled) /complete/search?...
```

```
(page load) --page loaded--
```

```
(new url) https://www.google.com/search?q=gi...
```

```
(xhr) ● GET 200 /async/bgasy?ei=hn...
```

```
(xhr) ● GET 204 /client_204?cs=2
```

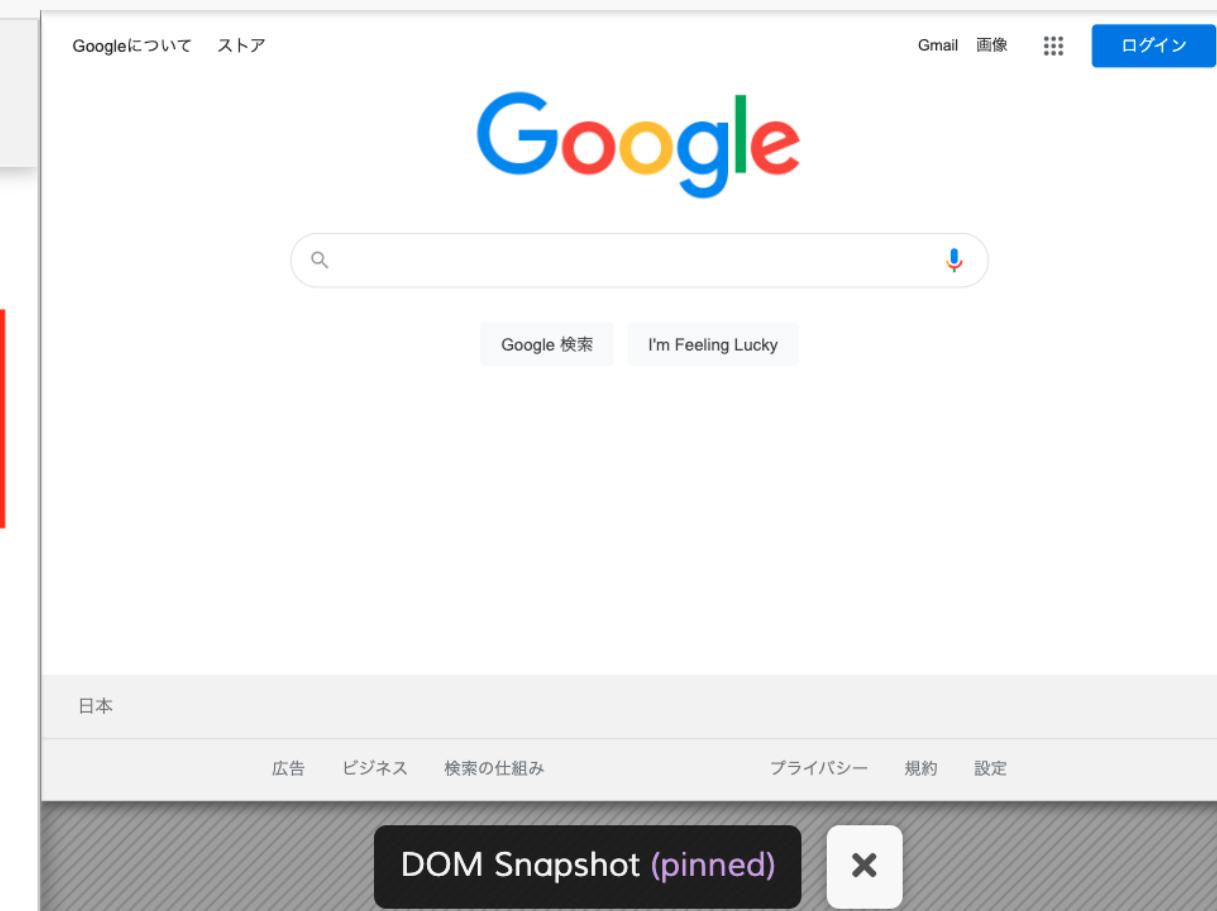
```
(xhr) ● GET --- /complete/search?q=githu...
```

```
(xhr) ● GET --- /complete/search?q&cp=0&...
```

```
3 get a[href="https://github.com/"]
```

```
4 - assert
```

```
expected <a> to have a length of 1
```



Cypress

TEST CODE

```
/// <reference types="cypress" />

context('Google', () => {
  beforeEach(() => {
    cy.visit('https://www.google.com/');
  });

  it('github finds https://github.com/', () => {
    cy.get('input.gLFyf.gsfi')
      .type('github{enter}');

    cy.get('a[href="https://github.com/"]')
      .should(($anchor) => {
        expect($anchor).to.have.length(1);
      });
  });
});
```

cypress/integration/examples/google.spec.js

Google

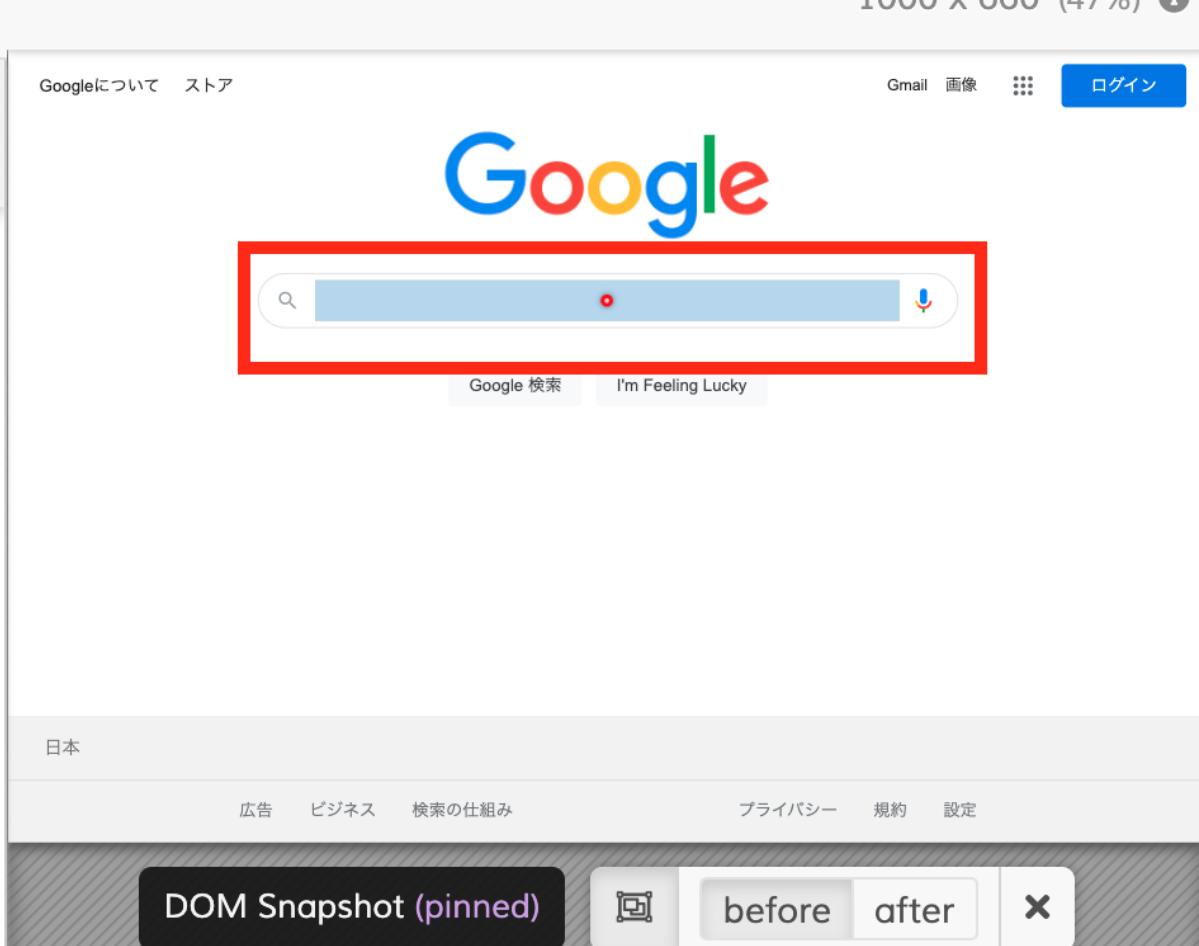
✓ github finds https://github.com/

▼ BEFORE EACH

1 visit
https://www.google.com/
(xhr)
● GET (canceled) /complete/search?...

▼ TEST BODY

1 get
input.gLFyf.gsfi
- type github{enter}
(xhr)
● GET (canceled) /complete/search?...
(page load) --page loaded--
(new url)
https://www.google.com/search?q=gi...
(xhr)
● GET 200 /async/bgasy?ei=hn...
(xhr)
● GET 204 /client_204?cs=2
(xhr)
● GET --- /complete/search?q=githu...
(xhr)
● GET --- /complete/search?q&cp=0&...
3 get
a[href="https://github.com/"]
4 - assert
expected <a> to have a length
of 1



Cypress

TEST CODE

```
/// <reference types="cypress" />

context('Google', () => {
  beforeEach(() => {
    cy.visit('https://www.google.com/');
  });

  it('github finds https://github.com/', () => {
    cy.get('input.gLFyf.gsfi')
      .type('github{enter}')

    cy.get('a[href="https://github.com/"]')
      .should(($anchor) => {
        expect($anchor).to.have.length(1);
      });
  });
});
```

cypress/integration/examples/google.spec.js

Google

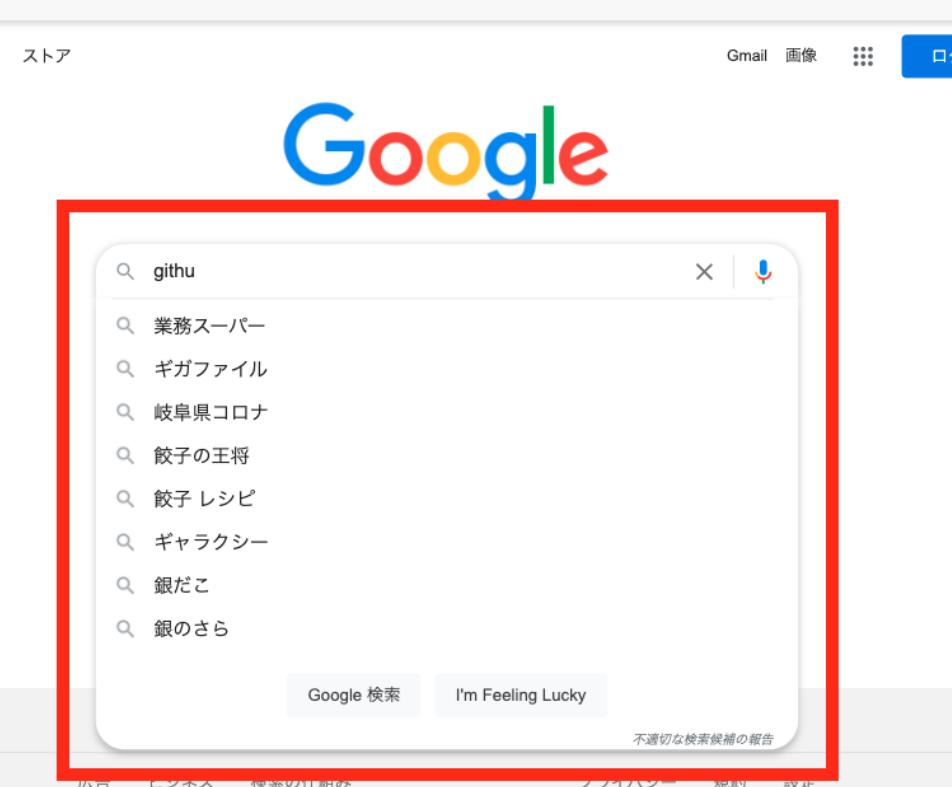
✓ github finds https://github.com/

▼ BEFORE EACH

1 visit
https://www.google.com/
(xhr)
● GET (canceled) /complete/search?...

▼ TEST BODY

1 get
input.gLFyf.gsfi
2 -type github{enter}
(xhr)
● GET (canceled) /complete/search?...
(page load) --page loaded--
(new url)
https://www.google.com/search?q=gi...
(xhr)
● GET 200 /async/bgasy?ei=hn...
(xhr)
● GET 204 /client_204?cs=2
(xhr)
● GET --- /complete/search?q=githu...
(xhr)
● GET --- /complete/search?q=cp=0&...
3 get
a[href="https://github.com/"]
4 - assert
expected <a> to have a length
of 1



DOM Snapshot (pinned)

request canceled

Cypress

TEST CODE

```
/// <reference types="cypress" />

context('Google', () => {
  beforeEach(() => {
    cy.visit('https://www.google.com/');
  });

  it('github finds https://github.com/', () => {
    cy.get('input.gLFyf.gsfi')
      .type('github{enter}')

    cy.get('a[href="https://github.com/"]')
      .should(($anchor) => {
        expect($anchor).to.have.length(1);
      });
  });
});
```

cypress/integration/examples/google.spec.js

Google

✓ github finds https://github.com/

▼ BEFORE EACH

1 visit
https://www.google.com/
(xhr)
● GET (canceled) /complete/search?...

▼ TEST BODY

1 get
input.gLFyf.gsfi
2 -type github{enter}
(xhr)
● GET (canceled) /complete/search?...
(page load) --page loaded--
(new url)
https://www.google.com/search?q=gi...
(xhr)
● GET 200 /async/bgasy?ei=hn...
(xhr)
● GET 204 /client_204?cs=2
(xhr)
● GET --- /complete/search?q=githu...
(xhr)
● GET --- /complete/search?q&cp=0&...

3 get
a[href="https://github.com/"]
4 - assert
expected <a> to have a length
of 1

GitHub: Where the world builds software · GitHub

GitHub is where over 65 million developers shape the future of software, together. Contribute to the open source community, manage your Git repositories, review code like a pro, track bugs and features, power your CI/CD and DevOps ...

<https://ja.wikipedia.org/wiki/GitHub>

GitHub - Wikipedia - ウィキペディア

GitHub (ギットハブ) は、ソフトウェア開発のプラットフォームであり、ソースコードをホスティングする。コードのバージョン管理システムにはGitを使用する。Ruby on RailsおよびErlangで記述されており、アメリカのカリフォルニア州 ...

運営者：GitHub, Inc
ユーザー数：2800万人（2018年6月）
言語：英語
開始：2008年4月

<https://techacademy.jp/> プログラミング, Git ▾

今さら聞けない！GitHubの使い方【超初心者向け】

今まで使ったことがない初心者向けに「GitHubの使い方」を解説した記事です。Gitを使う上で知っておきたい、ローカルリポジトリと ...

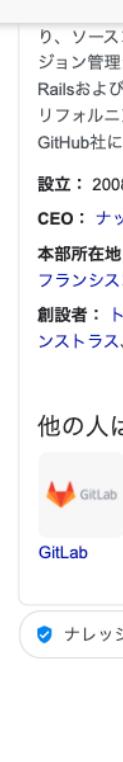
2020/03/31 · アップロード元: TechAcademy [テックアカデミー]

国 トップニュース

SSH越しでのGitHub操作にセキュリティキーが利用可能に-2021年内にパスワードを廃止へ

ZDNet Japan · 2日前

DOM Snapshot (pinned)



Cypress with Mattermost

Cypress with Mattermost

REFERENCE PROJECT

- すでに Cypress でテストが書かれているプロジェクトがあるので、それらを真似っこ
- [mattermost/mattermost-webapp](#)
 - Mattermost Webapp 側本体 (React/Redux)
- [mattermost/mattermost-incident-collaboration](#)
 - Mattermost プラグイン
- [mattermost/mattermost-plugin-apps](#)
 - Mattermost プラグイン

Cypress with Mattermost

IMPORTANT FILES

- Mattermost のテストを書く上で重要な要素

```
e2e/
├── cypress/
│   ├── fixtures/
│   │   └── example.json
│   ├── integration/
│   │   └── unread_specs.ts
│   ├── plugins/
│   │   └── index.ts
│   ├── support/
│   │   ├── index.d.ts
│   │   └── index.ts
│   └── tsconfig.json
└── cypress.json
    ├── package-lock.json
    └── package.json
```

Cypress with Mattermost

PLUGINS FILE

- ブラウザテストの裏で Node プロセスを実行可能に
 - 例: ブラウザテスト中、別のチャネルへの投稿を REST API 経由で実行

```
// mattermost-webapp/e2e/cypress/plugins/post_message_as.js
const axios = require('axios');

module.exports = async ({sender, message, channelId, rootId, createdAt = 0, baseUrl}) => {
  try {
    response = await axios({
      url: `${baseUrl}/api/v4/posts`, method: 'post',
      headers: { 'Content-Type': 'application/json', 'X-Requested-With': 'XMLHttpRequest', Cookie: cookieString},
      data: { channel_id: channelId, message, type: '', create_at: createdAt, parent_id: rootId, root_id: rootId},
    });
  ...
}
```

```
// mattermost-webapp/e2e/cypress/plugins/index.js
const postMessageAs = require('./post_message_as');

...
module.exports = (on, config) => {
  on('task', {..., postMessageAs, ...}); // cy.task('postMessageAs', ...)
  ...
}
```

Cypress with Mattermost

SUPPORT FILE

- Spec 実行前に実行されるファイル

```
// mattermost-webapp/e2e/cypress/support/task_command.js
Cypress.Commands.add('postMessageAs', ({sender, message, channelId, rootId, createdAt}) => {
  const baseUrl = Cypress.config('baseUrl');

  return cy.task(
    'postMessageAs',
    {sender, message, channelId, rootId, createdAt, baseUrl}
  ).then(({status, data}) => {
    ...
  });
});
```

```
// 使用例
cy.postMessageAs({
  sender: user,
  message: `test`,
  channelId: townsquareChannelId,
});
```

- 共通処理などを、各 Spec ファイルで`import`しなくても良くなる

Cypress with Mattermost

SUPPORT FILE

いくつかのAPI呼び出しをまとめた便利メソッドも用意できる

```
// mattermost-webapp/e2e/cypress/support/api/setup.js
Cypress.Commands.add(
  'apiInitSetup',
  ({
    teamPrefix = { name: 'team', displayName: 'Team' },
    channelPrefix = { name: 'channel', displayName: 'Channel' },
    ...
  } = {}) => {
  return cy.apiCreateTeam(teamPrefix.name, teamPrefix.displayName)
    .then(({ team }) => {
      // # Add public channel
      return cy.apiCreateChannel(team.id, channelPrefix.name, channelPrefix.displayName)
        .then(({ channel }) => {
          return cy.apiCreateUser({prefix: userPrefix || (promoteNewUserAsAdmin ? 'admin' : 'user')})
            .then(({ user }) => {
              if (promoteNewUserAsAdmin) {
                cy.apiPatchUserRoles(user.id, ['system_admin', 'system_user']);
              }
              return cy.apiAddUserToTeam(team.id, user.id).then(() => {

```

Cypress with Mattermost

DEPENDS ON PLUGINS/SUPPORTS OF MATTERMOST-WEBAPP

Mattermost WebAppの Plugins / Supports をimportして使うこともできそう

```
// e2e/cypress/plugins/index.ts
module.exports = require('mattermost-webapp/e2e/cypress/plugins');
```

```
// e2e/cypress/supports/index.ts
import 'mattermost-webapp/e2e/cypress/support'
```

ただし、mattermost-webappはNPM化されていないため...

- 依存ライブラリに何が必要なのかわからない
- supportsで読み込まれるconfig項目が分からぬ

```
// package.json
...
{
  "devDependencies": {
    ...
    "mattermost-webapp": "github:mattermost/mattermost-webapp#76c6005c24f1cf5a761912ebf60e41ae058b8c3",
    ...
  }
}
```

Cypress with Mattermost

CYPRESS.JSON | PACKAGE.JSON

```
// cypress.json
{
  "baseUrl": "http://localhost:8065",
  "viewportWidth": 1300,
  "defaultCommandTimeout": 20000,
  "taskTimeout": 20000,
  "pluginID": "com.mattermost.apps",
  "pluginFile": "com.mattermost.apps-0.1.0.tar.gz",
  "video": false,
  "ignoreTestFiles": "**/node_modules/**/*",
  "integrationFolder": "..",
  "testFiles": "**/*_spec.{js,ts}",
  "env": {
    "adminUsername": "admin",
    "adminPassword": "admin%%123",
    "ciBaseUrl": "",
    "dbClient": "mysql",
    "dbConnection": "mmuser:mostest@tcp(localhost:3306)/mattermost_test?charset...",
    "enableApplitools": false,
    ...
  }
}
```

- 指定しないと動かない要素がいっぱいあるけど、どれが必須か分からな...

Cypress with Mattermost

CYPRESS.JSON | PACKAGE.JSON

```
// package.json
...
"devDependencies": {
  "@testing-library/cypress": "^7.0.6",
  "@types/node": "^15.0.2",
  "@types/testing-library__cypress": "^5.0.8",
  "axios": "0.20.0",
  "cypress": "^7.2.0",
  "cypress-file-upload": "4.1.1",
  "cypress-plugin-tab": "1.0.5",
  "cypress-wait-until": "1.7.1",
  "deepmerge": "4.2.2",
  "lodash.mapkeys": "4.6.0",
  "lodash.xor": "4.5.0",
  "mattermost-redux": "5.33.1",
  "mattermost-webapp": "github:mattermost/mattermost-webapp#76c6005c24f1cfa5a761912ebf60e41ae058b8c3",
  "mochawesome": "6.1.1"
}
...
}
```

- 依存に追加しないと動かないモジュールがいっぱいあるけど、どれが必須か分からぬ...

Cypress with Mattermost

CYPRESS.JSON | PACKAGE.JSON

<https://github.com/mattermost/mattermost-redux>

Mattermost Redux is now located at mattermost/mattermost-webapp This repository is being left open for the time being while *we set up proper monorepo infrastructure to be able to release mattermost-redux from there*, but it will be archived once that has been set up.

New releases of mattermost-redux will be on hold while that is done, but feel free to continue using existing versions.

monorepo構成になってcypress周りのコードも利用しやすくなることを期待
(モジュールとして利用可能になるのはreduxだけかも...)

Testing Mattermost Plugin

Testing Mattermost Plugin

```
// e2e/cypress/integration/unread_spec.ts
describe('subscribe unreads post', () => {
  let testTeam, testChannel, testUser;
  before(() => {
    const newSettings = {
      PluginSettings: {
        Enable: true,
      },
      ServiceSettings: {
        EnableTesting: true,
        EnableDeveloper: true,
      },
    };
    cy.apiUpdateConfig(newSettings);
    cy.apiUploadAndEnablePlugin({
      id: 'com.github.kaakaa.mattermost-plugin-unreads',
      version: '0.1.0',
      // e2e/cypress/fixtures/com.github.kaakaa.mattermost-plugin-unreads-0.1.0.tar.gz
      filename: 'com.github.kaakaa.mattermost-plugin-unreads-0.1.0.tar.gz'
    });
    cy.apiInitSetup().then(({team, channel, user}) => {
      testTeam = team;
      testChannel = channel;
      testUser = user;
    })
  })
})
```

Testing Mattermost Plugin

```
...
beforeEach(() => {
  cy.apiAdminLogin();
  cy.visit(`/${testTeam.name}/channels/town-square`);
  // wait to render
  cy.get('#postListContent').should('be.visible');
  // Open RHS
  cy.get('.row.main i.icon-plugin-unreads').should('be.visible').and('exist').click();
});

it('show an unread message', () => {
  const message = '1: test message';
  cy.postMessageAs({sender: testUser, message: message, channelId: testChannel.id})
    .then(({id}) => {
      cy.get(`.plugin-unreads-channel__${testChannel.id}`).should('have.length', 1);
      cy.get(`.plugin-unreads-post__${id} .plugin-unreads-post__content p`).should('have.text', message);
    });
});
...

```

- e2e
 - cypress
 - integration
 - metrics_specs.js
 - rhs_specs.js
 - unwind_specs.js



An update (7.4.0) is available. [Learn more](#)



Performance Monitoring

Performance Monitoring

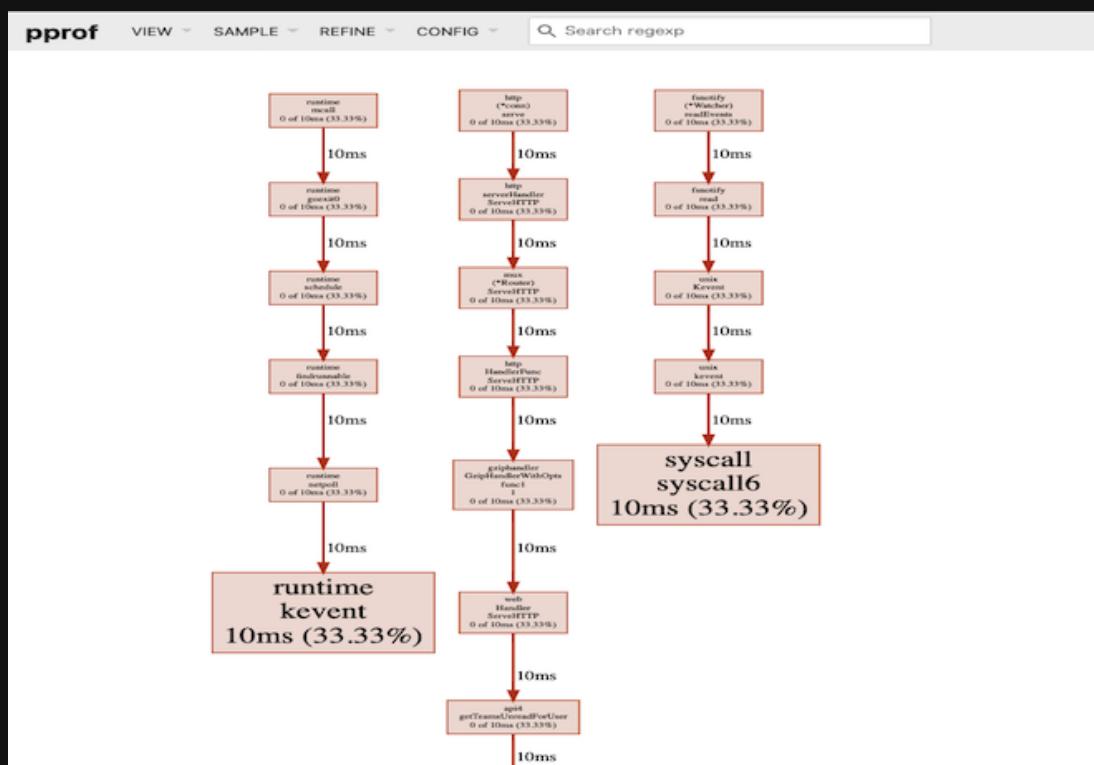
Mattermost Pluginのパフォーマンスを測れそうなもの

- **pprof (server)**
- **cAdvisor (container)**
- **Chrome DevTools Protocol (browser)**

Performance Monitoring

SERVER PROFILE

- pprof
 - Goプロセス (HTTPサーバー含む) のプロファイリングをするGo言語の機能
 - v5.34 (2021-04-14リリース)から、Team Editionでも利用できるようになった
 - <https://docs.mattermost.com/administration/changelog.html#release-v5-34-feature-release>
 - 今回のプラグインはServer側の機能追加をしないので使用しない
 - また、動的に推移を見れないのでpprof単体だと計測は難しい？



Performance Monitoring

CONTAINER PROFILE

- cAdvisor?
 - Mattermost開発環境は、コンテナを使って環境を構築する
 - ただ、Cypressテストで使われるコンテナはMySQLだけで、さらにcAdvisorではDisk I/Oは取れなさそうなので、あまり意味はないかも
- Mattermost ServerプロセスはただのGoプロセスとして実行されている


cAdvisor

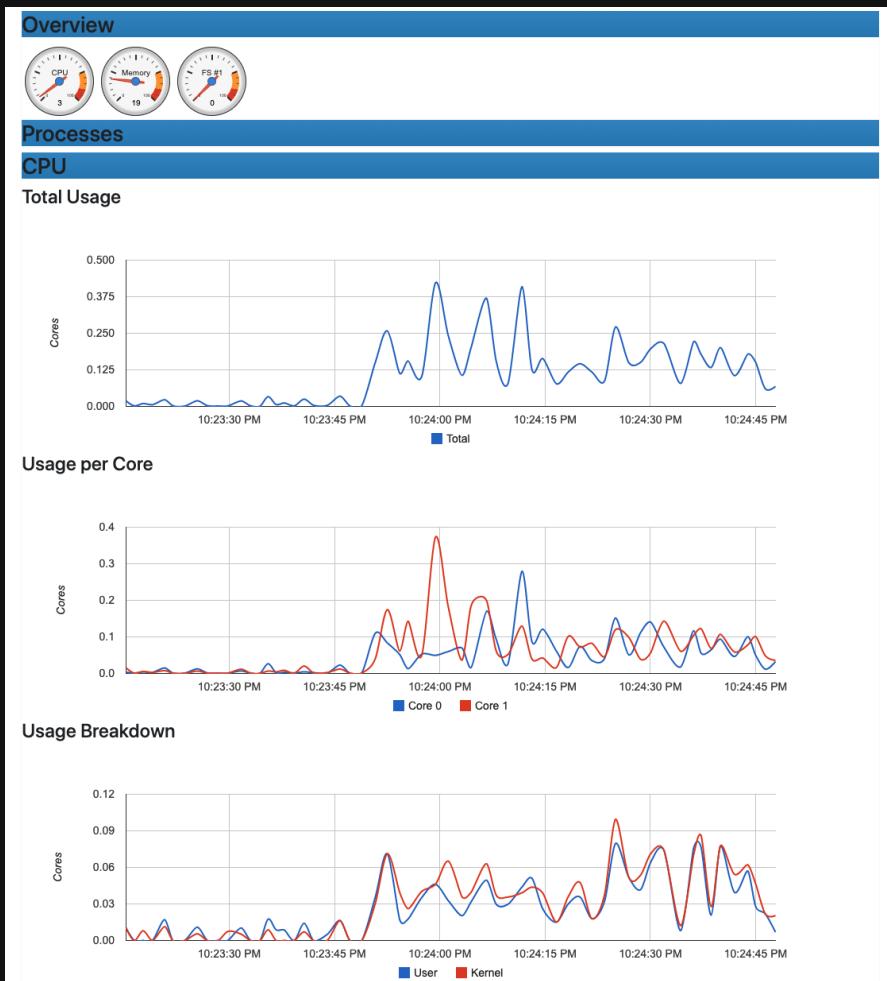
Docker Containers

Subcontainers

- mattermost-minio (/docker/1a946bff4a56c5661cd15cdc36d43beb899fb0a91aedd072afb972f2c17c10dc)
- mattermost-mysql (/docker/2887396ffafab51c49dc47f5cc1895018c165d5375f7ef5ac9e5f98cbf852909d)
- mattermost-postgres (/docker/aea7cae7f1cd8fc979474c794fc674c7facbf8564634dded57dc81bf6e109b9)
- cadvisor (/docker/959dcfa52fd2a8d6ff7aa75ffdcb3be12f0773fff8f2861e16890c31d0516762)
- mattermost-inbucket (/docker/589f683c01ae3d27625d8a6cd0fb66212ba34c9a6e688b9bb03f1055bc3ab7c9)

Driver Status

- Docker Version 20.10.2
- Docker API Version 1.41
- Kernel Version 4.19.121-linuxkit
- OS Version Docker Desktop
- Host Name docker-desktop
- Docker Root Directory /var/lib/docker
- Execution Driver
- Number of Images 104
- Number of Containers 18



Performance Monitoring

BROWSER PROFILE

■ Using Cypress and Google Lighthouse for Performance Testing | Applitools ?

- Lighthouseとはウェブページの品質改善の指針を「パフォーマンス」、「PWA」、「アクセシビリティ」、「ベストプラクティス」、「SEO」点でチェックしてくれるツール。

■ Lighthouseの計測結果を見ていく - Qiita

■ Mattermost本体に対して使うようなツール？

■ Cypress - Chrome dev tools profiling stats. · Issue #7894 · cypress-io/cypress

- You can access the **Chrome DevTools Protocol** like below if you need something more specific.

```
Cypress.automation("remote:debugger:protocol", {  
})
```

■ Devtools Protocol ?🤔

DevTools Protocolを使ってChromeを操作 & CSSカバレッジを取得する - Qiita

ご存知ですか? DevTools Protocol

ブラウザオートメーションといえば **Selenium** の名前を思い浮かべる方は多いと思います。 Selenium 2 からの実装である **WebDriver** は **W3C** で標準化が進められおり、ブラウザオートメーションの標準規格として存在しているものだと（少なくとも私は）捉えていました。

しかし近年リリースされた **Puppeteer** や **Cypress** などをみると WebDriver は利用されておらず、この DevTools Protocol が利用されています。 WebDriver と比較した特徴として以下が挙げられます。

- 動作が高速
- 安定
- 多機能

Chrome DevTools Protocol の API は以下ドキュメントが公開されており、

<https://chromedevtools.github.io/devtools-protocol/>

WebDriver では定義されていなかった DevTools の各種機能（JavaScriptデバッガ、プロファイラ、パフォーマンスツール、etc...）が利用可能なことが読み取れます。

Chrome (&Chromium) だけしか使えないのでは？と思うところですが、 WICGのリポジトリにリソースが集められており、各ブラウザでの実装状況もここから確認できます。

Performance Monitoring

CYPRESS WITH CHROME DEVTOOLS PROTOCOL

- Chrome DevTools Protocol - Performance domain

- - Performance.getMetrics: Retrieve current values of run-time metrics.

```
describe('metrics', () => {
  it('test', () => {
    Cypress.automation('remote:debugger:protocol', {
      command: 'Performance.getMetrics'
    }).then((val) => cy.log(val));
  })
})
```

```
{
  "metrics": [
    ...
    {"name": "JSHeapUsedSize", "value": 62996760},
    {"name": "JSHeapTotalSize", "value": 91262976}
  ]
}
```

- Chromeのメモリとか取れるけど、スナップショットなので推移を見る仕組みを作る必要がありそう
 - やってない

おまけ

GitHub Actions | Cypress Documentation

- - Basic Setup
 - Run the Cypress tests within our GitHub repository within Electron.
 - Testing in Chrome and Firefox with Cypress Docker Images
 - Using the Cypress Dashboard with GitHub Actions

Cypress with Mattermost in GitHub Actions?

- サービスコンテナについて - GitHub Docs
 - サービスコンテナは、ワークフロー中でアプリケーションをテストもしくは運用するのに必要になるかもしれないサービスをホストするための、シンプルでポータブルな方法を提供するDockerコンテナです。たとえば、ワークフローでデータベースやメモリキャッシュへのアクセスを必要とする結合テストを実行する必要があるかもしれません。
- mattermost/mattermost-preview

Wrap up

- 📝 Mattermostでも採用されているE2E Testing frameworkである **Cypress**について調べた
- 🚀 DL数は**Selenium**を超えており、WebアプリのE2Eテストフレームワークとしてはデファクト
- 💯 Mattermost本体ではCypressの **Plugin / Support** を利用してテストを簡潔(?)に記述している、
 - Mattermost本体のPlugin/Supportは(一応)ライブラリとして使用することもできる
- ⚙ Mattermost Plugin のパフォーマンスは **pprof(server)** と **Chrome Devtools Protocol** で測定可能？
 - どちらもスナップショットを取得するツールなので、推移を見る仕組みが必要

Impression

- Cypress デスクトップアプリが使いやすかった
 - ファイルを更新すると自動で再読み込み・テスト再実行される
- Mattermost Pluginのテストも、一度書き方を覚えればあとは数増やすだけの作業
 - mattermost-webappのcypressコードがモジュール化されると最高
- HTMLの**class/id**設計超重要
 - これがないとテストできない

終

↓ 正解: Vue Component ↓