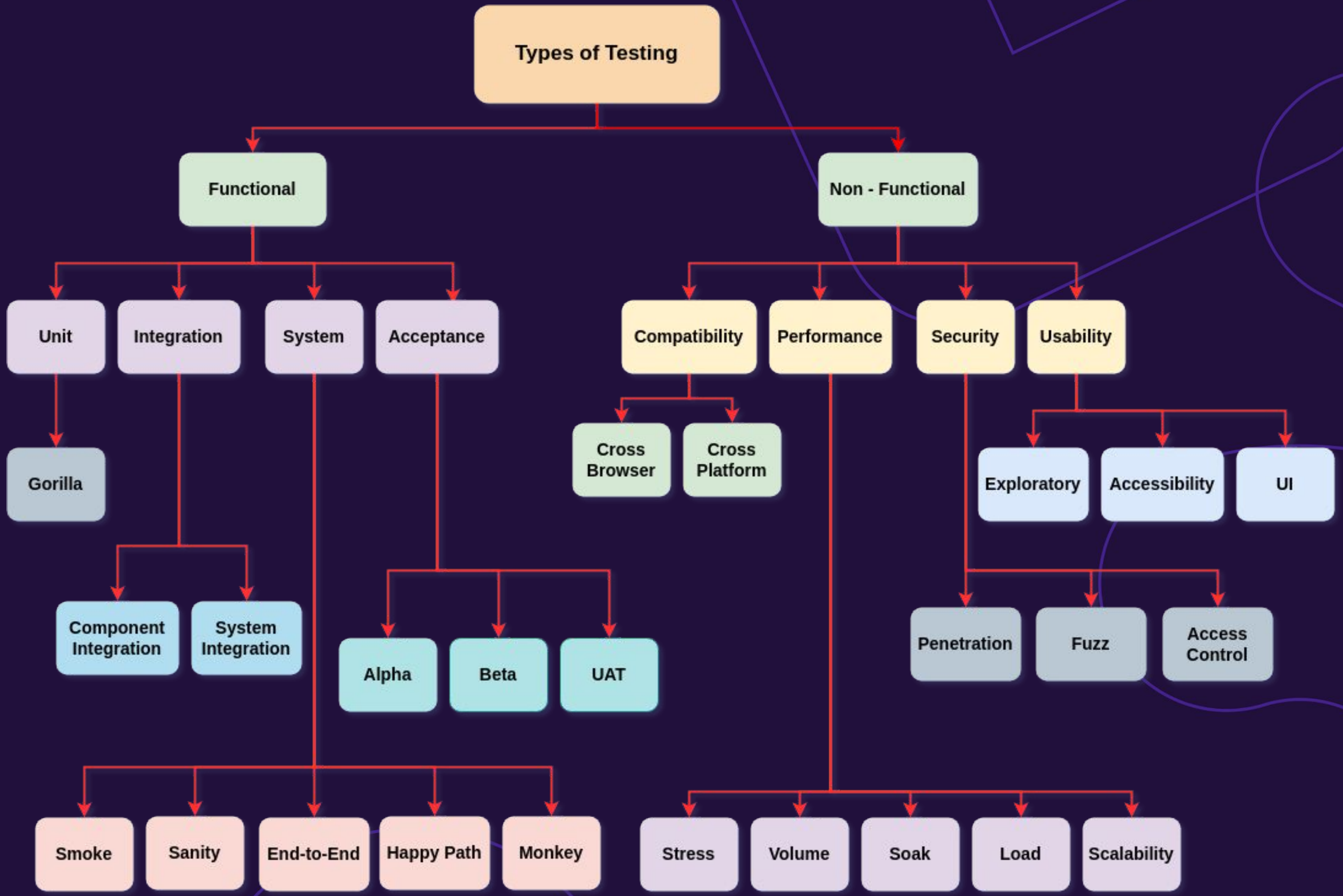
Firefox

Automated WebExtensions Testing

2024.11.20 Simeon Vincent — Developer Relations Engineer



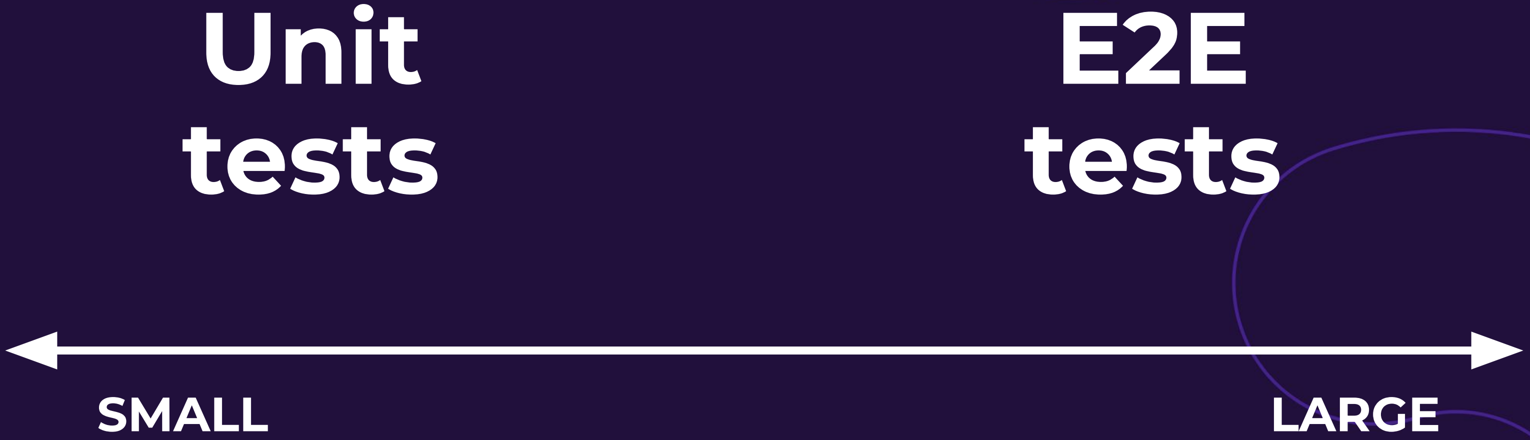
Types of Testing

- Functional
  - Unit
    - Gorilla
  - Integration
    - Component Integration
    - System Integration
  - System
    - Smoke
    - Sanity
    - End-to-End
    - Happy Path
    - Monkey
  - Acceptance
    - Alpha
    - Beta
    - UAT
- Non - Functional
  - Compatibility
    - Cross Browser
    - Cross Platform
  - Performance
    - Stress
    - Volume
    - Soak
    - Load
    - Scalability
  - Security
    - Penetration
    - Fuzz
    - Access Control
  - Usability
    - Exploratory
    - Accessibility
    - UI

Unit tests

End-to-end tests

Functional test continuum



Unit tests

E2E tests

SMALL

LARGE

Unit tests

A form of software testing by which isolated source code is tested to validate expected behavior

– Wikipedia

Photo by Chokniti Khongchum from Pexels:
https://www.pexels.com/photo/shallow-focus-photography-of-microscope-2280547/

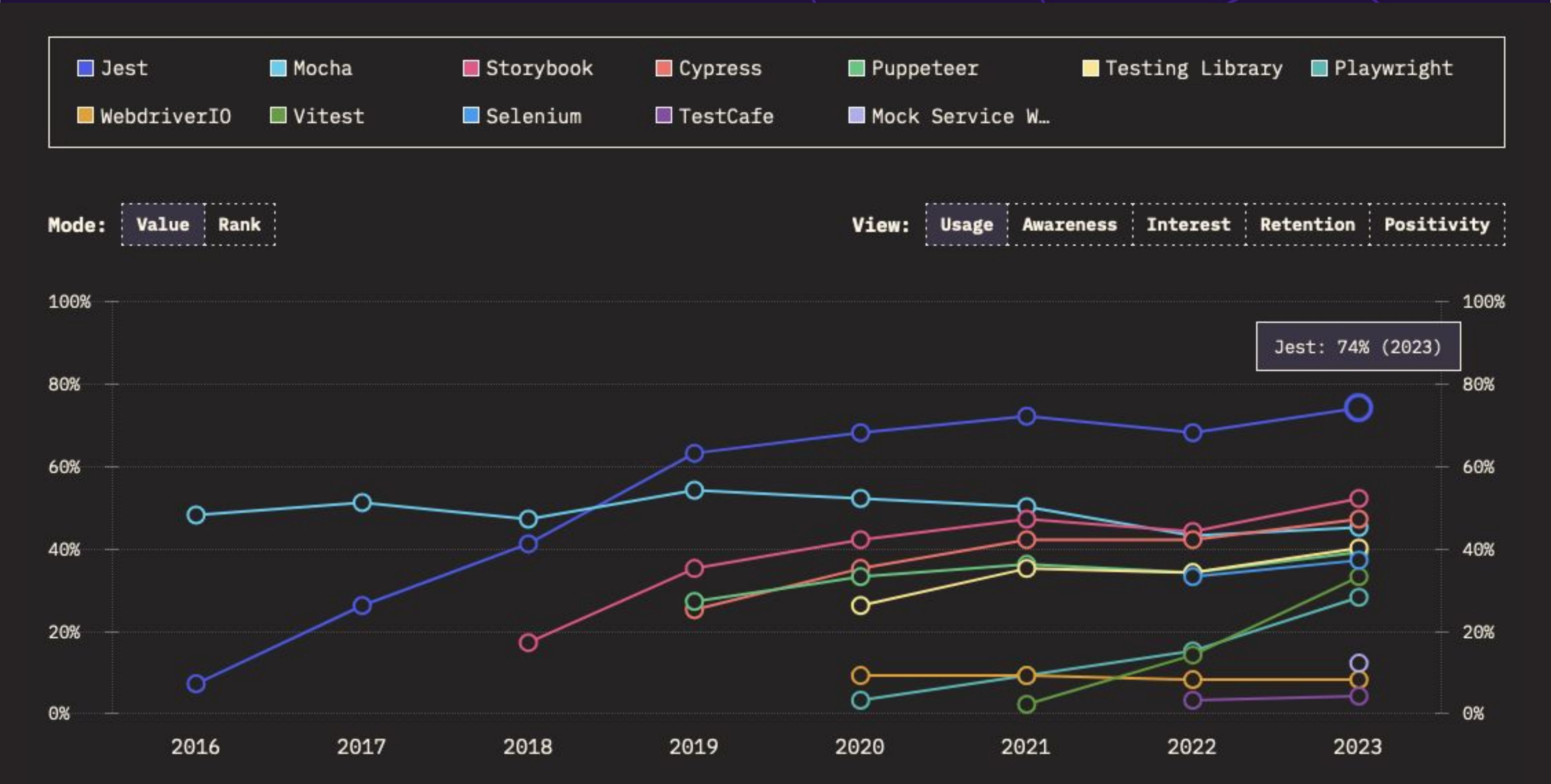
E2E tests

Testing conducted on a complete software system

– Wikipedia

Photo by RDNE Stock project from Pexels
https://www.pexels.com/photo/high-rise-buildings-in-los-angeles-8782721/

Unit test



https://2023.stateofjs.com/en-US/libraries/testing/

```
$ git clone https://github.com/dotproto/webext-test-setup.git
# Check out notes.md

$ git checkout start
```

https://github.com/dotproto/webext-test-setup.git

```
$ git clone https://github.com/dotproto/webext-test-setup.git
# Check out notes.md

$ git checkout start
```

https://github.com/dotproto/webext-test-setup.git

```
$ git clone https://github.com/dotproto/webext-test-setup.git
# Check out notes.md

$ git checkout start

$ npm install --save-dev jest
```

https://github.com/dotproto/webext-test-setup.git

```
$ git clone https://github.com/dotproto/webext-test-setup.git
# Check out notes.md

$ git checkout start

$ npm install --save-dev jest

$ vi package.json
```

https://github.com/dotproto/webext-test-setup.git

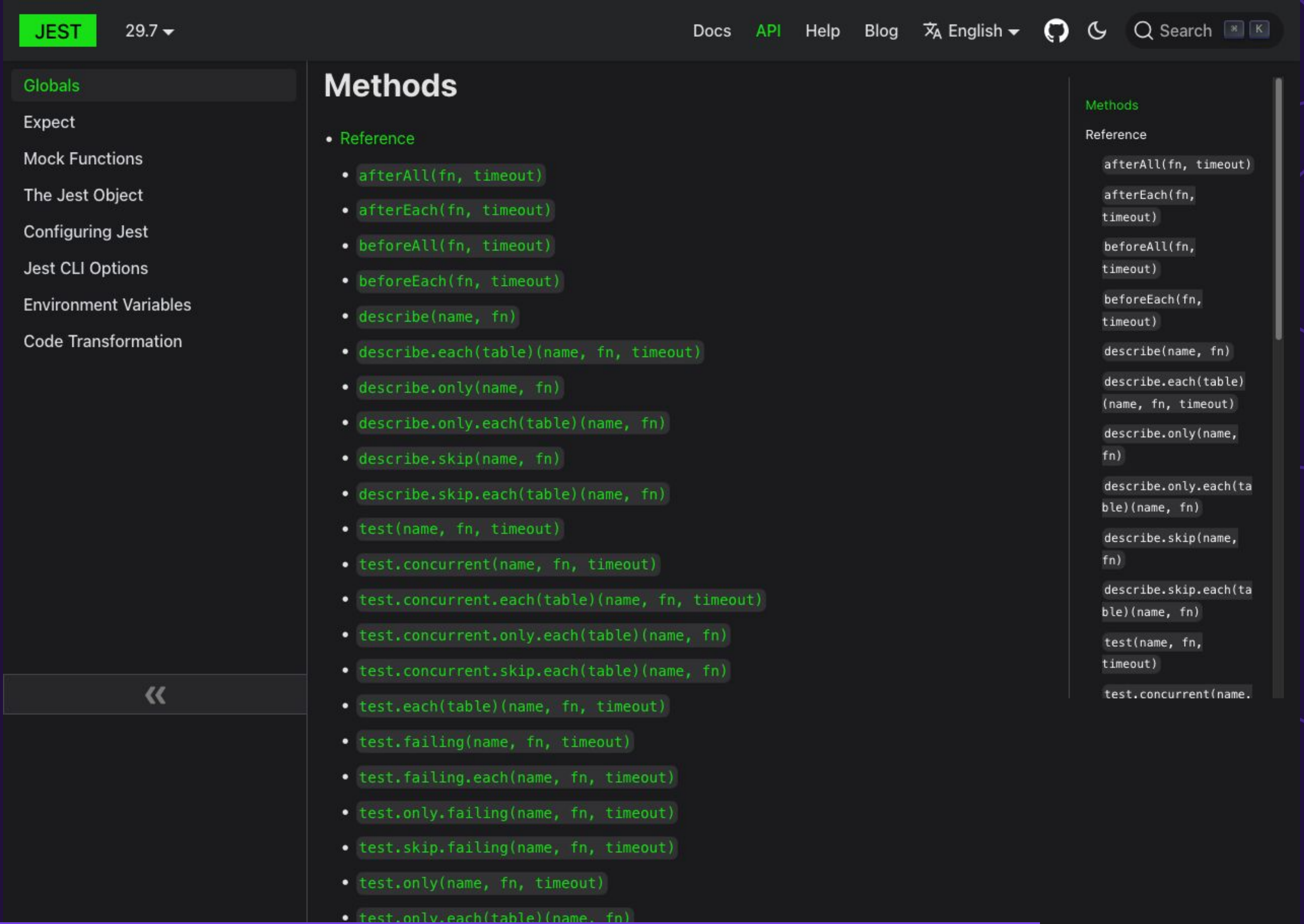
```
{
  "name": "store-collected-images",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {

    "build": "webpack --progress --color",
    "build:watch": "npm run build -- -w",
    "start": "npm run build && web-ext run -s extension/"
  },
  "author": "",
  "license": "MPL-2.0",
  "devDependencies": {
```

```
{
  "name": "store-collected-images",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest",
    "build": "webpack --progress --color",
    "build:watch": "npm run build -- -w",
    "start": "npm run build && web-ext run -s extension/"
  },
  "author": "",
  "license": "MPL-2.0",
  "devDependencies": {
```

```
$ npm run test
$ npm run test

> store-collected-images@1.0.0 test
> jest

No tests found, exiting with code 1
Run with `--passWithNoTests` to exit with code 0
In /Users/simeon/webext-test-setup
  12 files checked.
  testMatch: **/__tests__/**/*.[jt]s?(x), **/?(*.)+(spec|test)
  testPathIgnorePatterns: /node_modules/ - 12 matches
  testRegex:  - 0 matches
Pattern:  - 0 matches
```

```
$ vi src/utils/image-store.test.js
$ vi src/utils/image-store.test.js


import { loadStoredImages } from "./image-store.js";

test('should be a function', () => {
  expect(typeof loadStoredImages).toBe("function");
});
```

```
$ npm run test

> store-collected-images@1.0.0 test
> jest

 PASS  src/utils/image-store.test.js
✓ should be a function (1 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.335 s, estimated 1 s
Ran all test suites.
```



Methods

- Reference
  - `afterAll(fn, timeout)`
  - `afterEach(fn, timeout)`
  - `beforeAll(fn, timeout)`
  - `beforeEach(fn, timeout)`
  - `describe(name, fn)`
  - `describe.each(table)(name, fn, timeout)`
  - `describe.only(name, fn)`
  - `describe.only.each(table)(name, fn)`
  - `describe.skip(name, fn)`
  - `describe.skip.each(table)(name, fn)`
  - `test(name, fn, timeout)`
  - `test.concurrent(name, fn, timeout)`
  - `test.concurrent.each(table)(name, fn, timeout)`
  - `test.concurrent.only.each(table)(name, fn)`
  - `test.concurrent.skip.each(table)(name, fn)`
  - `test.each(table)(name, fn, timeout)`
  - `test.failing(name, fn, timeout)`
  - `test.failing.each(name, fn, timeout)`
  - `test.only.failing(name, fn, timeout)`
  - `test.skip.failing(name, fn, timeout)`
  - `test.only(name, fn, timeout)`
  - `test.only.each(table)(name, fn)`

Methods

Reference

`afterAll(fn, timeout)`

`afterEach(fn, timeout)`

`beforeAll(fn, timeout)`

`beforeEach(fn, timeout)`

`describe(name, fn)`

`describe.each(table)(name, fn, timeout)`

`describe.only(name, fn)`

`describe.only.each(table)(name, fn)`

`describe.skip(name, fn)`

`describe.skip.each(table)(name, fn)`

`test(name, fn, timeout)`

`test.concurrent(name,`

```
import { loadStoredImages } from "./image-store.js";

describe('loadStoredImages', () => {
  test('should be defined', () => {
    expect(typeof loadStoredImages).toBeDefined();
  });

  test('should be a function', () => {
    expect(typeof loadStoredImages).toBe("function");
  });
});
```

```
import { loadStoredImages } from "./image-store.js";

describe('loadStoredImages', () => {
  describe('argument handling', () => {
    //...
  });

  describe('return values', () => {
    //...
  });
});
```

```
import { loadStoredImages } from "./image-store.js";

describe('loadStoredImages', () => {
  describe.only('argument handling', () => {
    test.skip('Throws when called too few arguments', () => {
      // ...
    });

    test('Throws when called with too many arguments', () => {
      // ...
    });
  });
});
```

```
import { loadStoredImages } from "./image-store.js";

describe('loadStoredImages', () => {
  describe.only('argument handling', () => {
    test.skip('Throws when called too few arguments', () => {
      // ...
    });

    test('Throws when called with too many arguments', () => {
      // ...
    });
  });
});
```

Globals https://jestjs.io/docs/api

Basic structure

- describe()
- test() or it()

Setup

- beforeAll()
- afterAll()
- beforeEach()
- afterEach()

Focus management

- test.only()
- test.skip()
- test.failing()
- test.todo()

Also works for describe()

Expect https://jestjs.io/docs/expect

Equality

.toBe()   - Strict equality

.toEqual() - Object.is for primitives, deep equality for objs

.toHaveProperty() - Check specific (deep) properties

.toMatchObject() - Check if the object has some properties

Promises

.resolves - The promise eventually resolves

.rejects - The promise eventually rejects

Chain with other checks: expect(a).resolves.toBe(y)