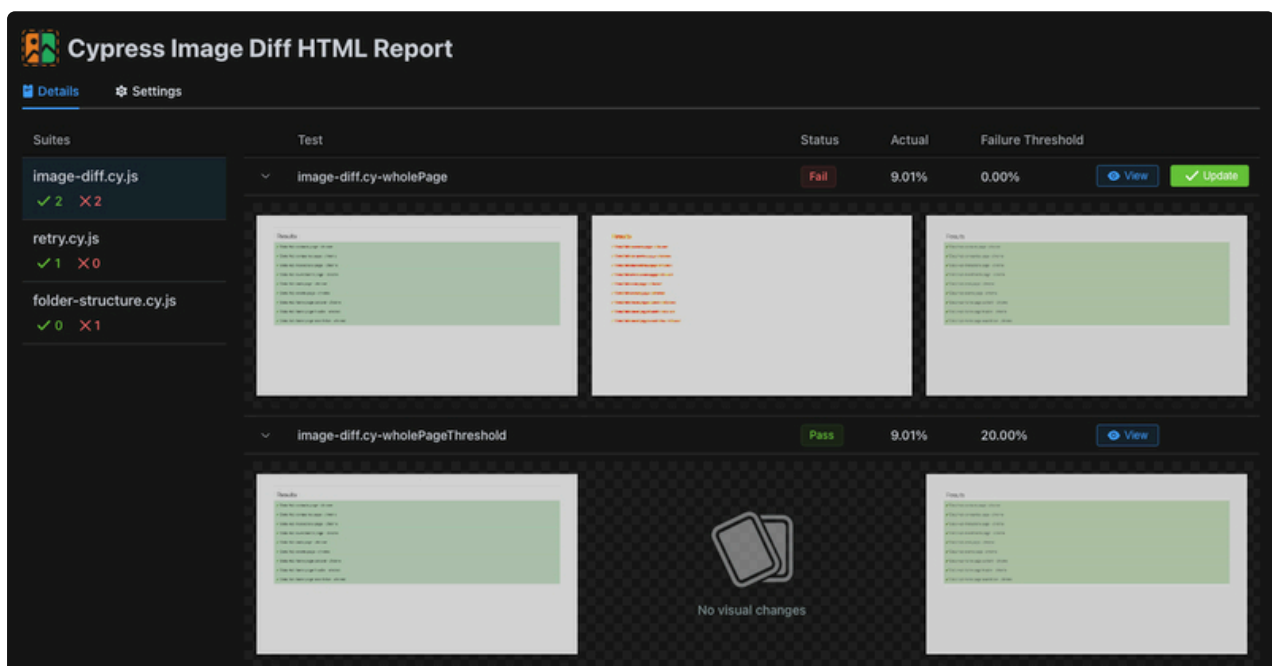


# **Cypress Image Diff Documentation**

# Migrate from v1 to v2

In a mission to provide a robust and easy-to-use visual regression tool for front-end development, **Cypress Image Diff** version 2 introduces [Cypress Image Diff HTML Report](#) as a separate npm package with extensive features and provides a new `compareSnapshot` arguments type.

## [New HTML Report](#)



We've separated out the core plugin and the HTML report (under the name of **Cypress Image Diff HTML Report**) with the aim of providing a beautiful interactive HTML report with a JSON file with all the details about your test suites, together with extensive features:

- Update baseline screenshots
- Toggle between different screenshot inspector modes: carousel, slider, and mirror
- Select your preferred colour scheme

Although **Cypress Image Diff HTML Report** is a standalone npm package, the [integration](#) with the core Cypress Image Diff is seamless, and they are expected to

work together out of the box without any further configuration.

*Note: The old report still continues to function in **Cypress Image Diff** version 2, but it is planned to be deprecated once the **Cypress Image Diff HTML Report** reaches its stable state.*

---

## New compareSnapshot arguments type

- compareSnapshot now could accept new arguments: cypressScreenshotOptions and exactName
- New global config option: CYPRESS\_SCREENSHOT\_OPTIONS

```
// cypress-image-diff v1.32.0 and older
cy.compareSnapshot('header')
cy.compareSnapshot('header', 0.2)
cy.compareSnapshot('header', 0.2, { limit: 6, delay: 1000 })

// cypress-image-diff v2.0.0 and newer
cy.compareSnapshot('header')
cy.compareSnapshot({
  name: 'header',
  testThreshold: 0.2
})
cy.compareSnapshot({
  name: 'header',
  testThreshold: 0.2,
  recurseOptions: { limit: 6, delay: 1000 }
})
cy.compareSnapshot({
  name: 'header',
  testThreshold: 0.2,
  recurseOptions: { limit: 6, delay: 1000 },
  // new custom cypress screenshot options
  cypressScreenshotOptions: {
    clip: { x: 0, y: 0, width: 100, height: 100 }
  }
  // new exactName option
  exactName: true
})
```

## Rename import path

```
❌ import getCompareSnapshotsPlugin from 'cypress-image-diff-js/dist/plugin';  
✅ import getCompareSnapshotsPlugin from 'cypress-image-diff-js/plugin';  
  
❌ import compareSnapshotCommand from 'cypress-image-diff-js/dist/command';  
✅ import compareSnapshotCommand from 'cypress-image-diff-js/command';
```

## Rename default screenshots and report folder

```
❌ cypress-visual-screenshots  
✅ cypress-image-diff-screenshots  
  
❌ cypress-visual-report  
✅ cypress-image-diff-html-report
```

## New config options

- make screenshots and report folder configurable

```
// cypress-image-diff.config.js  
const config = {  
  REPORT_DIR: 'html-report',  
  SCREENSHOTS_DIR: 'visual-screenshots'  
};  
  
module.exports = config;
```

*Note: `REPORT_DIR` and `SCREENSHOTS_DIR` only work for **Cypress Image Diff Html Report**, and this is intentional. Please consider using **Cypress Image Diff Html Report** for better support.*



# Getting started

Visual regression test with cypress

This tool was created to make visual regression as simple as possible, by exposing basic functions that allow you to view the difference between images. The wrapper uses [pixelmatch](#) which is simple and powerful and relies on [cypress](#) to take screenshots.

## Table of contents

- [Cypress integration](#)
- [Custom config file](#)
- [Reporting](#)
- [Client options](#)
- [Running tests](#)
- [Contributing](#)
- [Publishing](#)

## Plugin features

- [Screenshot of whole page](#)
- [Screenshot with increased pixel threshold tolerance](#)
- [Retry for an individual test](#)
- [Screenshot of an element](#)
- [Hide an element before taking a screenshot](#)

## Other

- [Guidelines for better testing results](#)
- [Update baseline images](#)
-

- [Folder structure](#)
- [Force resolution size](#)
- [Preserve cypress original screenshots](#)

## Getting started

Once you have setup cypress and followed [Cypress integration](#) you can start writing tests

### Writing a test

Create a spec file under cypress integration folder i.e

```
cypress/integration/specs/some-test-spec.js
```

Then use the cypress image diff command to take screenshots of pages or elements:

### Take screenshot and compare the whole page

```
describe('Visuals', () => {  
  it('should compare screenshot of the entire page', () => {  
    cy.visit('www.google.com')  
    cy.compareSnapshot('home-page')  
  })  
})
```

### Take screenshot with increased pixel threshold for an individual test

You can also make the comparison assertion more flexible by applying a higher threshold (default is 0):

```
describe('Visuals', () => {
  it('should compare screenshot of the entire page', () => {
    cy.visit('www.google.com')
    cy.compareSnapshot({
      name: 'home-page-with-threshold',
      testThreshold: 0.2
    })
  })
})
```

## Retry for an individual test

You can also retry the snapshot comparison by passing in an optional third parameter. It accepts the same options as [cypress-recurse](#).

```
describe('Visuals', () => {
  it('should compare screenshot of the entire page', () => {
    cy.visit('www.google.com')
    cy.compareSnapshot({
      name: 'home-page-with-threshold',
      testThreshold: 0,
      retryOptions: {
        limit: 5, // max number of retries
        delay: 500 // delay before next iteration, ms
      }
    })
  })
})
```

## Take screenshot and compare an element

```
describe('Visuals', () => {
  it('should compare screenshot from a given element', () => {
    cy.visit('www.google.com')
    cy.get('#report-header').compareSnapshot('search-bar-element')
  })
})
```

## Hiding an element before taking a screenshot



```
describe('Visuals', () => {
  it('should compare screenshot from a given element', () => {
    cy.visit('www.google.com')
    cy.get('#report-header').hideElement() // hideElement(false) to unhide
    cy.compareSnapshot('search-bar-element')
  })
})
```

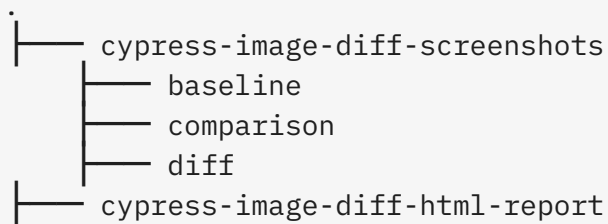
## Updating baseline images

If there are wanted changes to the application in test and if we need to update baseline images, you can follow the steps in [Client options](#) documentation to update the baselines.

Alternatively, you can delete the baseline image that you wish to be updated and rerun the tests, this will create a new baseline image with the updated image.

## Folder structure

This folder structure will be created by default at the root of your project where your `package.json` lives:



In some cases, you may want to modify the folder structure relative to the directory. To accomplish this you will need to set a value on the `ROOT_DIR` key value on the `cypress-image-diff.config.js` [custom config file](#). `ROOT_DIR` is a path relative to the current working directory.

## Force resolution size

In order to force the screenshot resolution when running a test you will need to set the following cypress config values in `cypress.json`:

```
{
  "viewportWidth": 1000, // Default value: 1280
  "viewportHeight": 660 // Default value: 720
}
```

## Preserving the original screenshot

All screenshots will be renamed and moved from the default screenshot location to a new screenshot folder structure. To preserve the screenshot in the original location, set the following values in `cypress.json`:

```
{
  "env": {
    "preserveOriginalScreenshot": true
  }
}
```

## Guidelines for better visual testing results

**Note:** This plugin internally uses `cypress.screenshot` command and due to its limitation on taking full page screenshots (<https://github.com/cypress-io/cypress/issues/2681>), you might encounter some weird results if you have `html`, `body` set to `100%`, or `scroll-behavior: smooth`, this workaround of resetting these CSS properties before invoking the command might help:

```
cy.get("html, body").invoke(
  "attr",
  "style",
  "height: auto; scroll-behavior: auto;"
);
```

Full page screenshots are generally not recommended over individual DOM elements, as stated in the [Cypress docs](#), so use them sparingly.

**Note:** Be aware that despite forcing a screenshot resolution to a particular height and width for a test, if this test is run on different infrastructure (i.e a 13"

Mac vs PC attached to a 30" monitor), the results will be different. So it's extremely important that you standardize where the tests will run, both locally and CI.

One way to handle this, is by running it with docker container or against BrowserStack or alike.

This project tests use a docker container to run the tests so it could be used as an example.

# Cypress integration

## Integration with Cypress

Follow the steps below to have access to the comparison command so you can build visual regression tests

## Setup

Install Cypress:

```
npm i -D cypress
```

Install the core package and the HTML report:

```
npm i -D cypress-image-diff-js cypress-image-diff-html-report
```

Then initialise Cypress if you don't have a project:

```
npx cypress open
```

Finally follow a suitable option for you below:

- [Typescript](#)
- [Javascript](#)
- [Cypress < v10](#)

Once the above is complete, you will be all set to [write a test](#)!

If you have any struggles with integration, please refer to [these examples](#).

# Typescript

## Cypress plugin

Import and initialise the Cypress image diff plugin:

```
// cypress.config.ts
import { defineConfig } from 'cypress';
import getCompareSnapshotsPlugin from 'cypress-image-diff-js/plugin';

export default defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      return getCompareSnapshotsPlugin(on, config);
    },
  },
});
```

## Cypress support

Import compareSnapshotCommand command:

```
// cypress/support/{scheme}.ts, where {scheme} defaults to e2e
import compareSnapshotCommand from 'cypress-image-diff-js/command';
// for Cypress v12.17.3 and older
import compareSnapshotCommand from 'cypress-image-diff-js';
compareSnapshotCommand();
```

## Troubleshooting

1. `>(0, command_1.default) is not a function`



Extend your `tsconfig.json` `compilerOptions` with:

```
"esModuleInterop": true,
```

in this case your `tsconfig.json` should look like this:

```
{
  "compilerOptions": {
    "target": "es5",
    "lib": ["es5", "dom"],
    "types": ["cypress", "node"],
    "esModuleInterop": true,
  },
  "include": ["**/*.ts"]
}
```

Note: You have to restart the Cypress application to reload the changes

# Javascript

## Cypress plugin

Import and initialise the Cypress image diff plugin:

```
// cypress.config.js
const { defineConfig } = require("cypress");
const getCompareSnapshotsPlugin = require('cypress-image-diff-js/plugin')

module.exports = defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      return getCompareSnapshotsPlugin(on, config);
    },
  },
});
```

## Cypress support

Import and add Cypress image command:

```
// cypress/support/{scheme}.js, where {scheme} defaults to e2e
const compareSnapshotCommand = require('cypress-image-diff-js/command');
// for Cypress v12.17.3 and older
const compareSnapshotCommand = require('cypress-image-diff-js');
compareSnapshotCommand();
```

# Browserstack

Reference: <https://www.browserstack.com/docs/automate/cypress>

## Setup

Install `browserstack-cypress-cli` package

- `yarn add browserstack-cypress-cli` OR `npm i browserstack-cypress-cli`

Run `browserstack-cypress-cli init`

- This will generate your `browserstack.json` file

Update the generated `browserstack.json` file with your Browserstack credentials and some configs

Required changes

- replace the `auth` credentials
- update `cypress_config_file` location, e.g. to `cypress.config.js`
- update `parallels` option, e.g. to `"1"`
- set `downloads` to your cypress-image-diff screenshot and report location:
  - it will be used later on when updating baseline images
  - for example:

```
"downloads": ["/cypress-image-diff-screenshots", "/cypress-image-c"]
```

- add all your test running dependencies to `npm_dependencies`

Recommended changes

- update the browsers section according to your preferences
  - reference: <https://www.browserstack.com/docs/automate/cypress/browsers-and-os>



# Running the tests

You can now run the tests with the `browserstack-cypress run` command

Reference: <https://www.browserstack.com/docs/automate/cypress#run-your-first-test>

## Artifacts

Upon running the `browserstack-cypress run` command with the `--sync` flag. It will sync the Browserstack build's artifacts to your local machine under `build_artifacts` folder.

Reference: <https://www.browserstack.com/docs/automate/cypress/custom-reports-build-artifacts>

## Updating baseline images

After running the tests with the previously mentioned `browserstack-cypress run` command you can simply copy the given comparison image from your local `build_artifacts/build_id/../../cypress-image-diff-screenshots/comparison` folder and move it to the `cypress-image-diff-screenshots/baseline` images folder. Which will serve as the new baseline images for any futures test runs.

## Creating browser + OS specific baseline images

In order to create browser specific baseline image we can use Cypress environmental variables.

For example:

```
describe('template spec', () => {
  it('passes', () => {
    cy.visit('../report-example.html')
    cy.contains('Data Hub home page search bar - chrome').should('be.visible')
    cy.compareSnapshot(`${Cypress.env('BROWSER')}-example`)
  })
})
```

When running the

```
CYPRESS_BROWSER="chrome@latest:OS X Ventura" browserstack-cypress run --
sync --browser "chrome@latest:OS X Ventura"
```

- It will override the browser configs in `browserstack.json` file
- Tests will generate browser/OS specific baselines image like  
`browserstack.spec.cy.ts-chrome@latest:OS X Ventura-example.png`

Drawback: You have to trigger a new browserstack build for each browser and OS combinations

## Example project setups

### Javascript

<https://github.com/tamasmagyar/cypress-image-diff-js-browserstack-integration>

### Typescript

<https://github.com/tamasmagyar/cypress-image-diff-js-browserstack-integration-typescript>

## Example browserstack.json

```
{
  "auth": {
    "username": "your_username",
    "access_key": "your_access_key"
  },
  "browsers": [
    {
      "browser": "chrome",
      "os": "Windows 10",
      "versions": [
        "latest",
        "latest-1"
      ]
    }
  ],
  "run_settings": {
    "cypress_config_file": "cypress.config.js",
    "project_name": "project-name",
    "build_name": "build-name",
    "exclude": [],
    "parallels": "1",
    "npm_dependencies": {
      "browserstack-cypress-cli": "^1.25.3",
      "cypress": "^13.2.0",
      "cypress-image-diff-js": "^1.30.1"
    },
    "package_config_options": {},
    "headless": true
  },
  "connection_settings": {
    "local": false,
    "local_identifier": null,
    "local_mode": null,
    "local_config_file": null
  },
  "disable_usage_reporting": false
}
```

# Cypress < v10

## Cypress plugin

Import and initialise the Cypress image diff plugin:

```
// cypress/plugin/index.js
module.exports = (on, config) => {
  const getCompareSnapshotsPlugin = require('cypress-image-diff-js/plugin')
  return getCompareSnapshotsPlugin(on, config)
};
```

## Cypress support

Import and add Cypress image command:

```
// cypress/support/index.js
require('./commands');

// cypress/support/commands.js
compareSnapshotCommand();
```

# cy.compareSnapshot command

All available options for compareSnapshot command

`cy.compareSnapshot` command could be use in a most basic way:

```
cy.compareSnapshot('header')
```

or in a more configurable way:

```
cy.compareSnapshot({
  name: 'header',
  testThreshold: 0.2
})
```

- Available options are:

Option	Type	Required/Default value	Description
<code>name</code>	<code>string</code>	required	The name of the snapshots that will be generated
<code>testThreshold</code>	<code>number</code>	<code>0</code>	A number between <code>0</code> and <code>1</code> that represents the allowed percentage of pixels that can be different between the two snapshots
<code>retryOptions</code>	<code>Partial&lt;RecurseDefaults&gt;</code>	<code>{ limit:1 }</code>	Config objects passed to

Option	Type	Required/Default value	Description
			<a href="#">cypress-recurse</a>
<code>exactName</code>	<code>boolean</code>	<code>false</code>	If set to <code>true</code> , will use the given name as it is without transforming it to <code>[spec_file_name]-[name]</code>
<code>cypressScreenshotOptions</code>	<code>Partial&lt;Cypress.ScreenshotOptions &amp; Cypress.Loggable &amp; Cypress.Timeoutable&gt;</code>	<code>undefined</code>	options object to change the default behavior of <code>cy.screenshot()</code>

# Custom config file

## Options for custom config file

If you'd like to take advantages of additional features, you will need to set up the custom config file.

Create a file called `cypress-image-diff.config.js`. This should live along side `cypress.config.js`, in the root of the directory.

If your project is written in ESM module, then you would need to use `.cjs` instead of `.js` extension.

```
// cypress-image-diff.config.js
// cypress-image-diff.config.cjs
const config = {
  ROOT_DIR: 'custom-folder-name',
};

module.exports = config;
```

**Note:** In order to make this custom config values effective, remember to return `getCompareSnapshotsPlugin` instance inside function `setupNodeEvents`:

```
export default defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      return getCompareSnapshotsPlugin(on, config);
    },
  },
});
```

Currently supported values in the custom config file:

- [ROOT\\_DIR](#)
- [FAILURE\\_THRESHOLD](#)
- [RETRY\\_OPTIONS](#)
- [FAIL\\_ON\\_MISSING\\_BASELINE](#)

- [COMPARISON\\_OPTIONS](#)
- [JSON\\_REPORT](#)
- [CYPRESS\\_SCREENSHOT\\_OPTIONS](#)
- [REPORT\\_DIR](#)
- [SCREENSHOTS\\_DIR](#)



# ROOT\_DIR

The ROOT\_DIR value should be relative to the root of the directory.

⚠ Default value: `''`

```
// cypress-image-diff.config.js
const config = {
  ROOT_DIR: 'visual-test/custom-folder-name',
};

module.exports = config;
```

Output directory:

```

├── visual-test
│   ├── custom-folder-name
│   │   ├── cypress-visual-screenshots
│   │   │   ├── baseline
│   │   │   ├── comparison
│   │   │   └── diff
│   │   └── cypress-visual-report
```

# FAILURE\_THRESHOLD

Create a file called `cypress-image-diff.config.js`. This should live along side `cypress.config.js`, in the root of the directory.



Default value: `0`

```
// cypress-image-diff.config.js
const config = {
  FAILURE_THRESHOLD: 0.1,
};

module.exports = config;
```

This pass all tests as long as the difference between the baseline and comparison is not greater than 10%

# RETRY\_OPTIONS

⚙️ Default value: `{}`

```
// cypress-image-diff.config.js
const config = {
  RETRY_OPTIONS: {
    log: true,
    limit: 50, // max number of iterations
    timeout: 30000, // time limit in ms
    delay: 300, // delay before next iteration, ms
  },
};

module.exports = config;
```

You can find all available options here: [retry options](#)

# FAIL\_ON\_MISSING\_BASELINE

Boolean to determine whether to fail a test if its baseline doesn't exist, default to false.

🔧 Default value: `false`

```
// cypress-image-diff.config.js
const config = {
  FAIL_ON_MISSING_BASELINE: true,
};

module.exports = config;
```

This is beneficial if you constantly get engineers pushing new along with their baselines. Failing the test in CI or locally.

# COMPARISON\_OPTIONS

Custom options passed to pixelmatch, default to `{ threshold: 0.1 }`. Please note that the `COMPARISON_OPTIONS.threshold` is different from the `FAILURE_THRESHOLD`

- `COMPARISON_OPTIONS.threshold` : is the failure threshold for every pixel comparison
- `FAILURE_THRESHOLD` : is the failure threshold for the whole comparison

🔧 Default value: `{ threshold: 0.1 }`

```
// cypress-image-diff.config.js
const config = {
  COMPARISON_OPTIONS: { threshold: 0.2 },
};

module.exports = config;
```

You can verify all available and updated options here: [pixelmatch options](#)

# JSON\_REPORT

Below are the options available to generate the JSON report

- **FILENAME:** custom name for the json report file, default to `report_[datetime].json` in which `[datetime]` will be replaced with a timestamp. (ie. `report_29-08-2023_233219.json` )
- **OVERWRITE:** set to false if you don't want to overwrite existing report files, default to true. If a duplicate filename is found, the report will be saved with a counter digit added to the filename. (ie. `custom_report_name_1.json` )



Default value: `{ FILENAME: 'report_[datetime].json', OVERWRITE: true }`

```
// cypress-image-diff.config.js
const config = {
  JSON_REPORT: {
    FILENAME: 'cypress_visual_report',
    OVERWRITE: false,
  },
};

module.exports = config;
```

# CYPRESS\_SCREENSHOT\_OPTIONS

We use the `Cypress.screenshot` command under the hood, and `CYPRESS_SCREENSHOT_OPTIONS` is an options object to change the default behaviour of the `Cypress.screenshot` command.

✖ Default value: `{}`

See all available options [here](#).

# REPORT\_DIR

The `REPORT_DIR` specifies where the generated JSON report lives. It's relative to the [ROOT\\_DIR](#).

🔧 Default value: `cypress-image-diff-html-report`

```
// cypress-image-diff.config.js
const config = {
  ROOT_DIR: 'visual-test/custom-folder-name',
  REPORT_DIR: 'html-report',
};

module.exports = config;
```

Output directory:

```
├── visual-test
│   ├── custom-folder-name
│       └── html-report
```



# SCREENSHOTS\_DIR

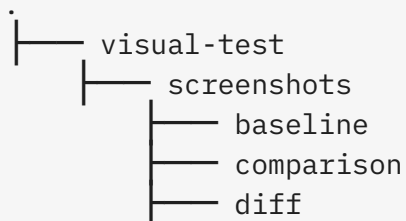
The `SCREENSHOTS_DIR` specifies where all the screenshots are saved. It's relative to the [ROOT\\_DIR](#).

✖ Default value: `cypress-image-diff-screenshots`

```
// cypress-image-diff.config.js
const config = {
  ROOT_DIR: 'visual-test',
  SCREENSHOTS_DIR: 'screenshots'
};

module.exports = config;
```

Output directory:



# Reporting

Reports are usually a great way to help visualize the test results. These are available reports that you could generate after every test run:

- [JSON Report](#)
- [Cypress Image Diff HTML Report](#)
- [Legacy HTML Report](#)

# JSON report

If you want to generate your custom report, generate a report json file by passing `JSON_REPORT` in the [custom config file](#), and build your own HTML file from this json.

Example JSON could be found at the bottom of this page.

## Folder structure

When a JSON report is generated, it will be created in the following default folder:

```
├── cypress-image-diff-report
│   └── report_[datetime].json
```

Report folders can be customized via the [REPORT\\_DIR](#) config option.

Example JSON:

```
{
  "total": 6,
  "totalPassed": 3,
  "totalFailed": 3,
  "totalSuites": 3,
  "suites": [
    {
      "name": "image-diff.cy.js",
      "path": "cypress/specs/image-diff.cy.js",
      "tests": [
        {
          "status": "fail",
          "name": "image-diff.cy-wholePage",
          "percentage": 0.0900966398590363,
          "failureThreshold": 0,
          "specPath": "cypress/specs/image-diff.cy.js",
          "specFilename": "image-diff.cy.js",
          "baselinePath": "cypress-visual-screenshots/baseline/image-diff",
          "diffPath": "cypress-visual-screenshots/diff/image-diff.cy-whol",
          "comparisonPath": "cypress-visual-screenshots/comparison/image-
        },
        {
          "status": "pass",
          "name": "image-diff.cy-wholePageThreshold",
          "percentage": 0.0900966398590363,
          "failureThreshold": 0.2,
          "specPath": "cypress/specs/image-diff.cy.js",
          "specFilename": "image-diff.cy.js",
          "baselinePath": "cypress-visual-screenshots/baseline/image-diff",
          "diffPath": "",
          "comparisonPath": "cypress-visual-screenshots/comparison/image-
        },
        {
          "status": "pass",
          "name": "image-diff.cy-element",
          "percentage": 0,
          "failureThreshold": 0,
          "specPath": "cypress/specs/image-diff.cy.js",
          "specFilename": "image-diff.cy.js",
          "baselinePath": "cypress-visual-screenshots/baseline/image-diff",
          "diffPath": "",
          "comparisonPath": "cypress-visual-screenshots/comparison/image-
        },
        {
          "status": "fail",
          "name": "image-diff.cy-hideElement",
          "percentage": 0.0900966398590363,
          "failureThreshold": 0.
        }
      ]
    }
  ]
}
```

```

        "specPath": "cypress/specs/image-diff.cy.js",
        "specFilename": "image-diff.cy.js",
        "baselinePath": "cypress-visual-screenshots/baseline/image-diff",
        "diffPath": "cypress-visual-screenshots/diff/image-diff.cy-hide",
        "comparisonPath": "cypress-visual-screenshots/comparison/image-
    }
]
},
{
    "name": "retry.cy.js",
    "path": "cypress/specs/retry.cy.js",
    "tests": [
        {
            "status": "pass",
            "name": "retry.cy-retry",
            "percentage": 0,
            "failureThreshold": 0,
            "specPath": "cypress/specs/retry.cy.js",
            "specFilename": "retry.cy.js",
            "baselinePath": "cypress-visual-screenshots/baseline/retry.cy-r",
            "diffPath": "",
            "comparisonPath": "cypress-visual-screenshots/comparison/retry.
        }
    ]
},
{
    "name": "folder-structure.cy.js",
    "path": "cypress/specs/folder-structure-test/folder-structure.cy.js",
    "tests": [
        {
            "status": "fail",
            "name": "folder-structure.cy-wholePage",
            "percentage": 0.0900966398590363,
            "failureThreshold": 0,
            "specPath": "cypress/specs/folder-structure-test/folder-structu",
            "specFilename": "folder-structure.cy.js",
            "baselinePath": "cypress-visual-screenshots/baseline/folder-str",
            "diffPath": "cypress-visual-screenshots/diff/folder-structure.c",
            "comparisonPath": "cypress-visual-screenshots/comparison/folder
        }
    ]
}
],
"startedAt": "2023-08-25T09:53:41.477Z",
"endedAt": "2023-08-25T09:54:00.875Z",
"duration": 15318,
"browserName": "chrome",
"browserVersion": "116.0.5845.110",

```

```
"cypressVersion": "10.8.0"
}
```

# Cypress Image Diff HTML Report

From **Cypress Image Diff** version 2, **Cypress Image Diff HTML Report** is the recommended HTML report over the [Legacy HTML report](#). **Cypress Image Diff HTML Report** generates a beautiful HTML report out of a [JSON file](#), and offers extensive features:

- Update baseline screenshots
- Toggle between different screenshot inspector modes: carousel, slider, and mirror
- Select your preferred colour scheme

Once you've done the [integration part](#), these quick command lines will be helpful:

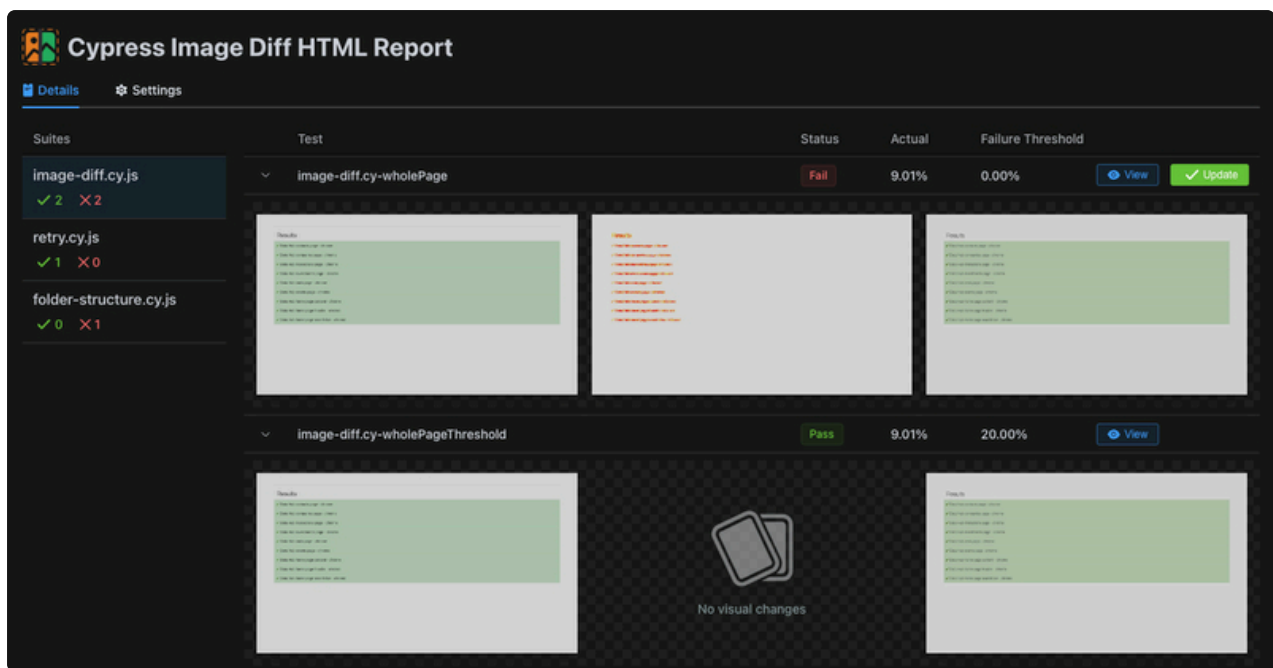
- `cypress-image-diff-html-report start` : to serve the HTML report out of a generated JSON file in an interactive mode, where you can update the baseline screenshots.
- `cypress-image-diff-html-report generate` : to generate and write to disc the HTML report in case you just want to view the static report.

Or you could integrate the report programmatically:

```
import { start } from 'cypress-image-diff-html-report'

;(async () => {
  await start({
    configFile: 'custom.config.js',
    serverPort: 3000
    // ...
  })
})()
```

[See here](#) for more details about how to use Cypress Image Diff HTML Report.



If you run the `generate` command, this following folder structure will be expected:

```
.
├── cypress-image-diff-html-report
│   ├── cypress-image-diff-html-report.html
│   └── report_[datetime].json
```

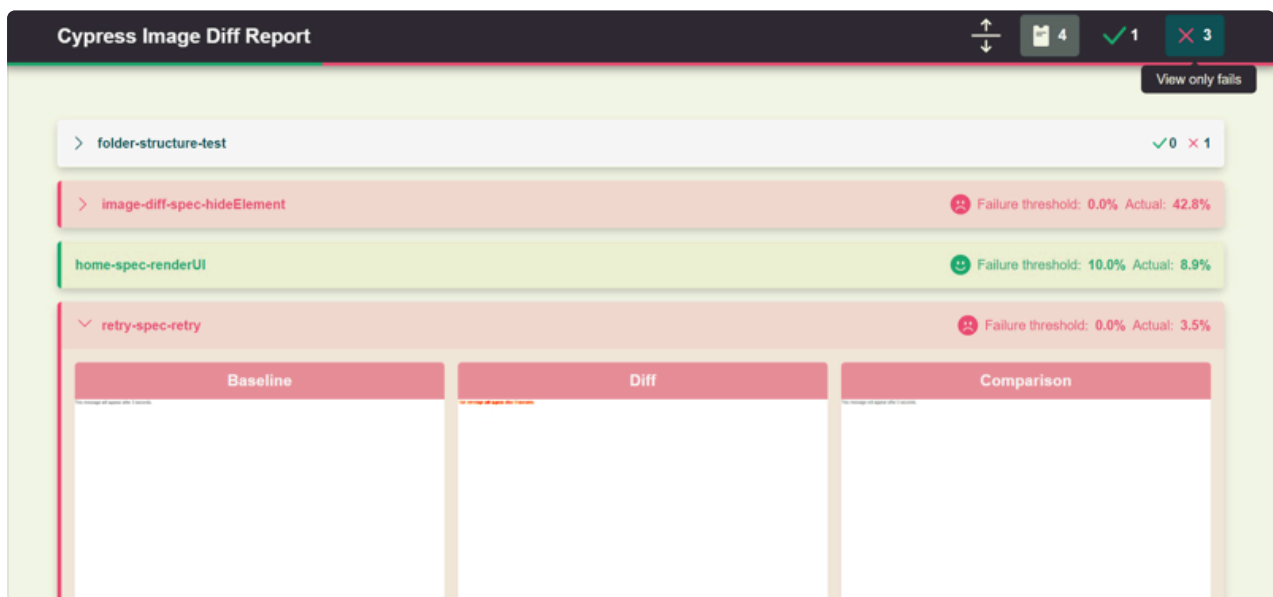
# Legacy HTML Report

Legacy HTML Report from version 1 continues to function in version 2, but is planned to be deprecated soon. Please consider using the [Cypress Image Diff HTML Report](#).

For some reasons, you still want to use the legacy HTML report, add the following `after` hook:

```
// cypress/support/index.js for Cypress versions below 10
// cypress/support/{scheme}.js for Cypress versions 10 and above, where {
after(() => {
  cy.task('generateReport')
})
```

The report will look something like:



*Note: Baseline, comparison and diff images will only be added to the report for failing tests.*

Legacy HTML report will be created following folder:

```
└─ cypress-image-diff-html-report
```



*Note: Report folder name for legacy HTML report can't be customized via `REPORT_DIR`, it's hardcoded as `cypress-image-diff-html-report` .*

# Client options

## Update baseline client option

In order to reduce manual work a cli option is available to copy over the comparison images into the baseline folder when there is a test failure.

### Update all baseline images for failing tests

Notice that you should run this command after the test suite runs. The below command will only update baseline images that have a diff image, which basically means a test failure.

```
$ cypress-image-diff -u
```

It's important that you ensure the comparison image is the correct representation of the page under test as it will be copied over to the baseline.

### Update a list of baseline images

This functionality is yet to be developed.

# Running tests

Run lint, unit and e2e tests on the repository

## Tests

Before you run any of the test command below ensure to build the image:

```
make build
```

## Linting

```
make lint-test
```

## Unit tests

```
make unit-test
```

## E2E tests

```
make e2e-test
```

## Please notice

It's important that you run the tests in the container as it will have a single resolution setup for everyone.

If the tests are executed locally, depending on your screen resolution the results can differ.

# Contributing

## Guidelines to contribute to this repository

When contributing to this repository, please first discuss the change you wish to make via issue, email, or any other method with the owners of this repository before making a change.

Please note we have a code of conduct, please follow it in all your interactions with the project.

## Commits

When committing please always use the following pattern for you messages (scope, body and footer are optional):

```
<type>(<scope>): <subject>
<BLANK LINE>
<body>
<BLANK LINE>
<footer>
```

Commit message types (tags)	Release type	Example
<b>feat:</b> A new feature	Minor release (0.1.0)	feat: Add "Investment project" activity card
<b>fix:</b> A bug fix	Patch release (0.0.1)	fix: Remove default activity card
<b>docs:</b> Documentation only changes	None	docs(README): Add testing instructions
<b>style:</b> Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)	None	style: Use tabs instead of spaces

Commit message types (tags)	Release type	Example
<b>refactor:</b> A code change that neither fixes a bug nor adds a feature	None	refactor: Add missing props validation to ActivityFeed
<b>perf:</b> A code change that improves performance	None	perf: Improve rendering speed of ActivityFeed
<b>test:</b> Adding missing or correcting existing tests	None	test: Add integration tests to ActivityFeedCard
<b>build</b> Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)	None	build: Update webpack config
<b>ci</b> Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs)	None	ci: Update CircleCI config
<b>chore:</b> Changes to the build process		

To create a major/breaking (1.0.0) release, please add **BREAKING CHANGE** to the commit message body with some explanation, example message:

```
fix: Update flaky screenshot function.
BREAKING CHANGE: Update third party driver object.
Optionally add more info in the second line of your message.
```

## Pull Request Process

1. Ensure any install or build dependencies are removed before the end of the layer when doing a build.
2. Update the README.md with details of changes to the interface, this includes new environment variables, exposed ports, useful file locations and container parameters.

3. You may merge the Pull Request in once you have the sign-off of the maintainer, or if you do not have permission to do that, you may request the reviewer to merge it for you.

## Code of Conduct

### Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

### Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

## **Our Responsibilities**

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

## **Scope**

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

## **Enforcement**

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [INSERT EMAIL ADDRESS]. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

## **Attribution**

This Code of Conduct is adapted from the [Contributor Covenant](http://contributor-covenant.org/version/1/4), version 1.4, available at <http://contributor-covenant.org/version/1/4>



# Publishing

Building and publishing

Commits to `main` automatically build and publish a new version of the package.

## Publising a new version

1. Create a PR with commits using one of the prefixes mentioned in CONTRIBUTING.
2. Have your PR reviewed and merged.