



A background image of a space shuttle launch. The shuttle is ascending vertically, leaving a massive, billowing plume of white smoke and fire. The launch pad structure is visible on the left, and a tall, thin tower stands on the right. The sky is a deep, dark blue.

Working **Well**: The Future of Web Testing

| Trent Willis @trentmwillis
QUnit Project Lead

NETFLIX

Testing Gives Us Confidence That Software Works

Testing **Should** Give Us Confidence
That Software Works **Well**

Works Well:
Performance
Visual Correctness
Code Necessity

chrome

@trentmwillis #assertjs

chrome **--headless**

Chrome DevTools Protocol

@trentmwillis #assertjs

chrome **--headless**
+
Chrome DevTools Protocol
=
New Possibilities

Puppeteer

@trentmwillis #assertjs

```
const browser = await puppeteer.launch();

const page = await browser.newPage();
await page.goto('https://example.com');

await page.content();
await page.screenshot({ path: 'screenshot.png' });

await browser.close();
```

Performance

@trentmwillis #assertjs

macro-benchmarks

@trentmwillis #assertjs

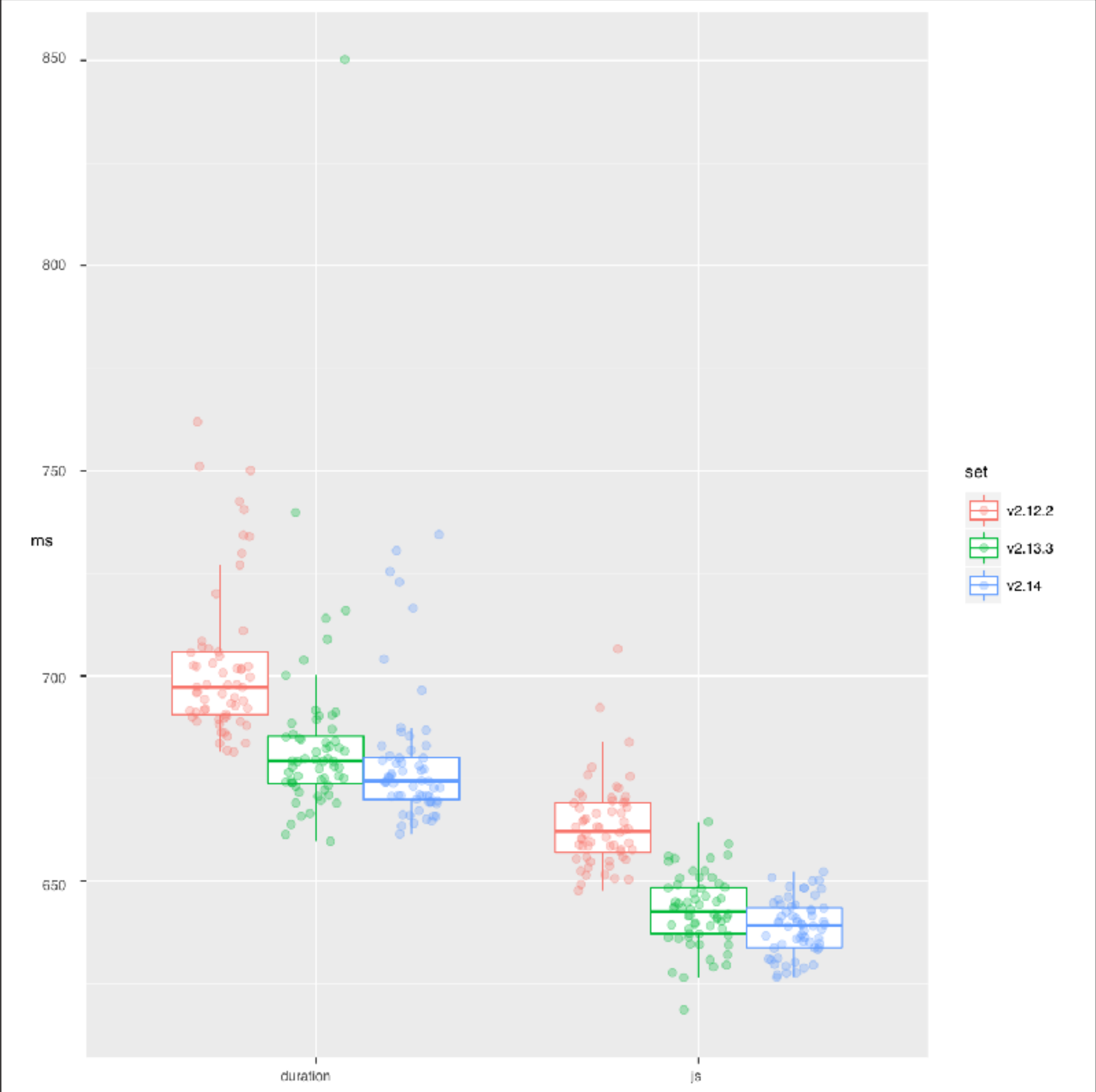
```
const pageMetrics = await page.metrics();  
const pageTimings = await page.evaluate(() => {  
  return performance.getEntries();  
});
```

```
await page.tracing.start({path: 'trace.json'});  
// do something  
await page.tracing.stop();
```

```
await page.tracing.start({path: 'trace.json'});  
// do something  
await page.tracing.stop();  
// repeat until statistically significant
```

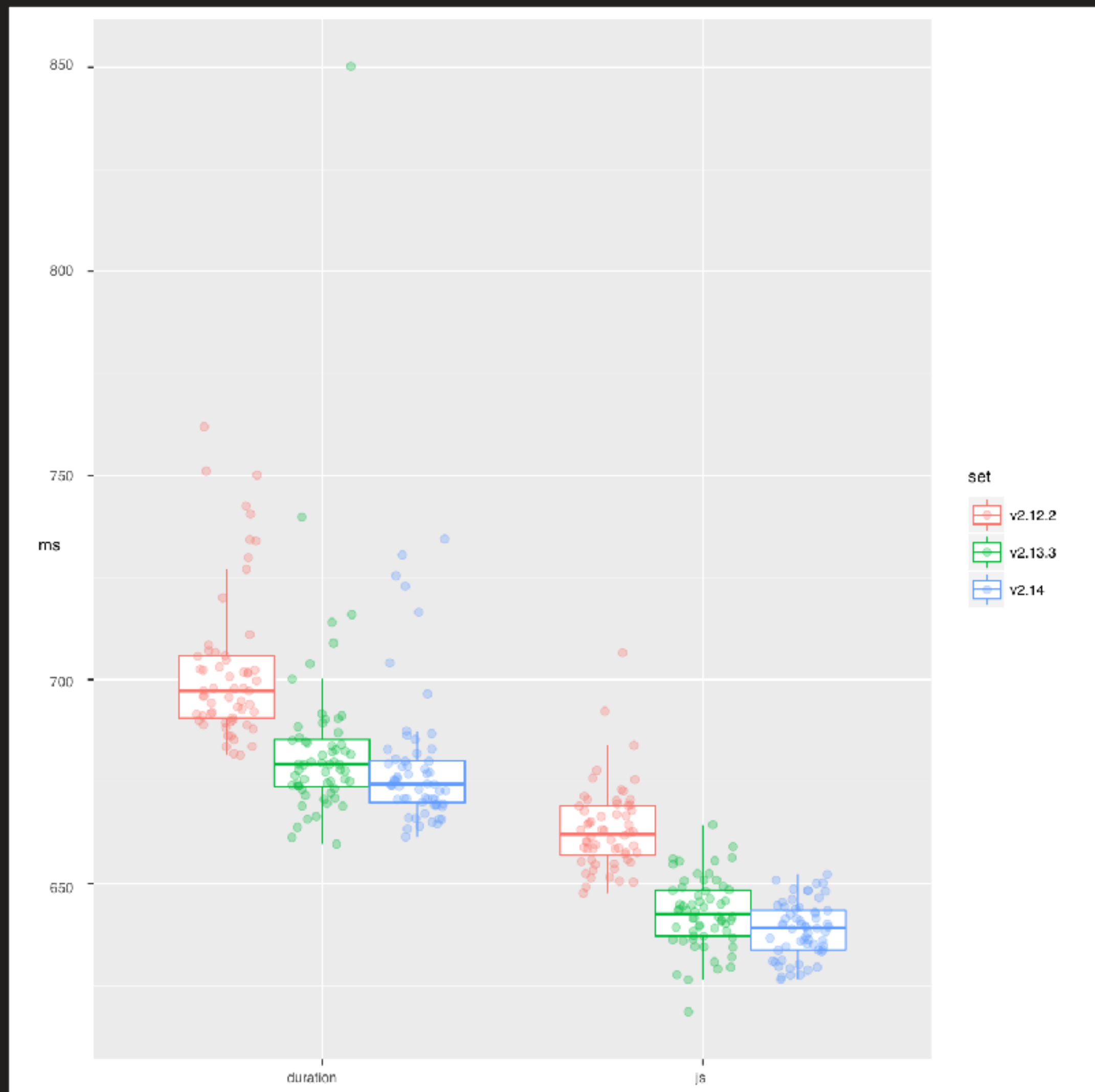


```
// mock network for reduce variance
await page.tracing.start({path: 'trace.json'});
// do something
await page.tracing.stop();
// repeat until statistically significant
```



@trentmwillis #assertjs

ember-macro-benchmark



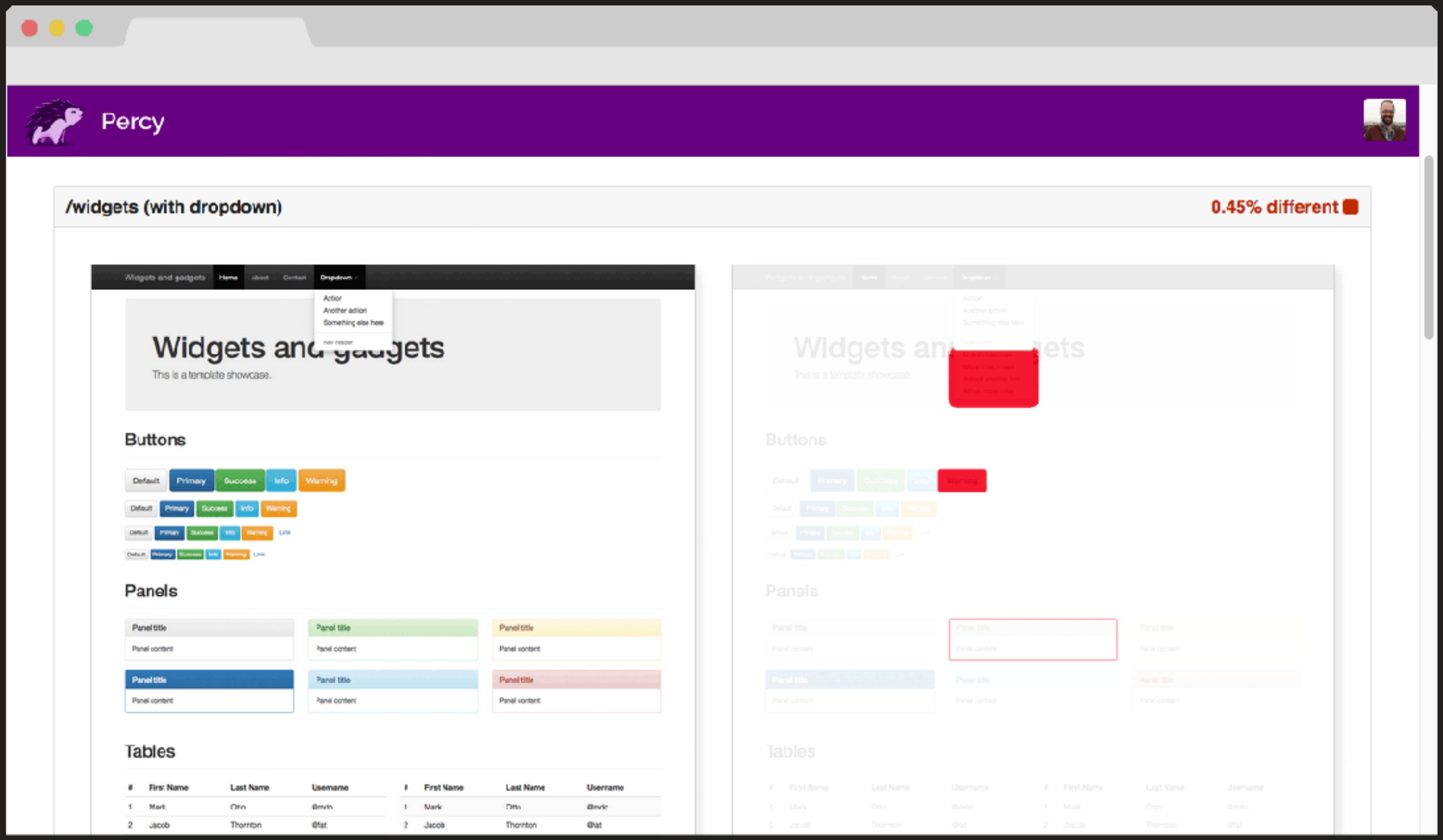
Visual Correctness

@trentmwillis #assertjs

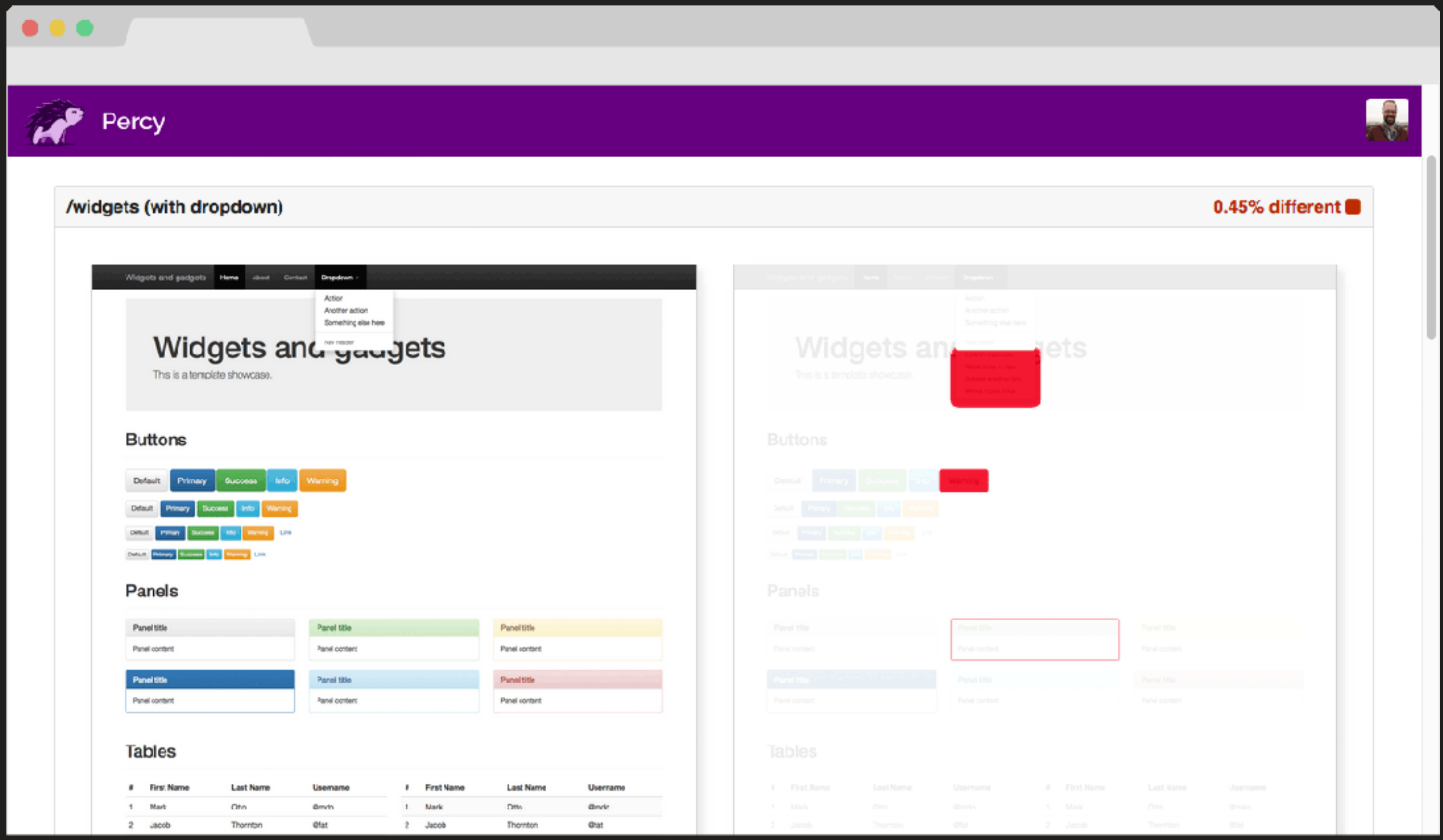
Automatic Visual Diffing
With Puppeteer
@notwaldorf

```
await page.screenshot({ path: 'screenshot.png' });
```

```
const oldImg = readImage('screenshot.png');
const newImg = readImage('screenshot-new.png');
const diff = new PNG({
  width: oldImg.width,
  height: oldImg.height,
});
const numDiffPixels = pixelmatch(
  oldImg.data,
  newImg.data,
  diff.data,
  oldImg.width,
  oldImg.height,
  { threshold: 0.1 },
);
assert.equal(numDiffPixels, 0);
```



@trentmwillis #assertjs








percy.io

Code **Necessity**

@trentmwillis #assertjs

Code Coverage

@trentmwillis #assertjs

URL	Type	Total Bytes	Unused Bytes	
/rs=ACT90oFcuqeAlZKgN_i7p0gUVf5FJbWmk	JS	415 893	193 290 46.5 %	
https://www.google.com/	CSS+JS	292 747	188 288 64.3 %	
/rs=AA2YrTvsJ4NG6EEJaw_kUXxLJ4pi7Db7G	JS	140 749	100 089 71.1 %	
https://apis.google.com/... /cb=gapi.loaded_0	JS	138 750	89 990 64.9 %	
ht/rs=ACT90oE78GyTaignaLd10ShHriearn9i4g	JS	125 995	76 665 60.8 %	

```
await page.coverage.startJSCoverage();  
// do something  
const coverage = await page.coverage.stopJSCoverage();
```

```
const percentOldCodeUsed = calculateUsage(oldCoverage);  
const percentNewCodeUsed = calculateUsage(newCoverage);  
assert.ok(percentNewCodeUsed < percentOldCodeUsed);
```

Works Well:
Performance
Visual Correctness
Code Necessity

QUnit-In-Browser

@trentmwillis #assertjs


```
test('quality check for home page', {
  url: `localhost:8080`,
  measurePerformance: true,
  enableScreenshots: true,
  measureCodeUsage: true,
}, async function(assert) {
  assert.equal(document.body.innerText, 'Hello world!');
  await takeScreenshot('home');
});
```

The image shows a VS Code editor window with the following components:

- EXPLORER:** Shows a file tree with folders for 'OPEN EDITORS' and 'QUNIT-IN-BROWSER'. The 'tests' folder is expanded, showing files like '_config.yml', '.eslintrc.json', '.gitignore', '.travis.yml', 'CHANGELOG.md', 'index.js', 'LICENSE', 'package-lock.json', 'package.json', 'README.md', and 'RELEASE.md'.
- EDITOR:** Displays the content of 'demo.js' with the following code:

```
3
4  const QUnit = require('qunit');
5  const server = require('./fixtures/server');
6  QUnit.test.inBrowser = require('../index');
7
8  QUnit.module('Assert(js) Demo');
9
10 QUnit.test.inBrowser('it works!', {
11   url: 'http://localhost:3000',
12   measurePerformance: true,
13   enableScreenshots: true,
14   measureCodeUsage: true,
15   server,
16 }, async function(assert) {
17
18   assert.equal(document.body.innerText, 'Hello world from a server!');
19   await takeScreenshot('home'); // eslint-disable-line no-undef
20
21 });
22
```
- TERMINAL:** Shows the command `nfml-twillisZEG:qunit-in-browser twillis$ qunit tests/demo.js` being executed.
- STATUS BAR:** At the bottom, it shows 'Ln 4, Col 32', 'Spaces: 2', 'UTF-8', 'LF', and 'JavaScript'.

@trentmwillis #assertjs

Testing **Should** Give Us Confidence
That Software Works **Well**



**The Future Is
Performant**



**The Future Is
Visually Correct**



**The Future Is
Necessary**



The Future Is Bright

| Trent Willis @trentmwillis
| QUnit Project Lead

NETFLIX