

Testing IMAIOS

GHOST INSPECTOR PRO / CON

After long research and a few discussions about our infrastructure,

I wanted to investigate if using a IDE might be a better idea. Since Ghost is based on a UI mechanism where most of the processes are done virtually through another distant server hosting multiple versions of different browsers it is a viable option to use the platform. However, Ghost Inspector lacks in the domain of fast based changes in the code and portability.

?

CODE Vs. GHOST

1. Text manipulation
2. Folder Hierarchy : network
3. Duplication of code for different modules in our site.

viewer and modules

TEST NAME	STATUS	SCREENSHOT	SCHEDULE	TESTED	More	Run Entire Suite
Cross-reff-Dog/Dog-Anatagy-Illustrations	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/07/19 4:09 PM	<input checked="" type="checkbox"/> API Access	<input type="button" value="Create New Test"/>
Cross-reff-Dog/Dog-Brain-MRI	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/07/19 4:10 PM		
Cross-reff-Head-and-Neck/Brain-MRI-in-axial-slices	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/07/19 4:09 PM		
Cross-reff-Head-and-Neck/Brain-diagrams	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/07/19 4:08 PM		
Cross-reff-Head-and-Neck/Head-CT	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/07/19 4:08 PM		
dropdown-compare-check	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/18/19 9:59 AM		
subscription.in.viewer	PASSING	<input type="radio"/> PASSING	Suite Schedule	06/18/19 9:59 AM		
viewer.maxSize.check	PASSING	<input type="radio"/> DISABLED	Suite Schedule	06/18/19 9:59 AM		

Renata from Ghost Inspector

> To: Gregoire Cousin

Friday 14 Jun at 18:58

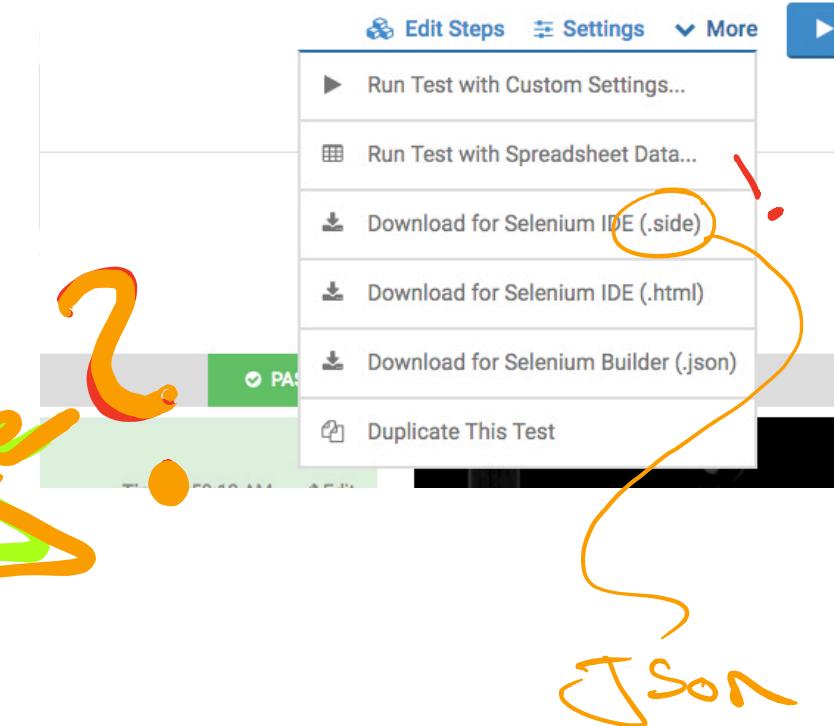
Hi there,

Thanks for reaching out. Unfortunately you will no be able to convert tests coded in JavaScript into HTML Selenium format because the JavaScript code is essentially freeform, it's not a set format, it supports a whole range of things that the HTML format does not (ex: loops), and could be written using various different frameworks.

To be perfectly honest with you, if you're already coding your tests using JavaScript, I'd recommend just using them in that format and instead subscribing to a CI service (ex: CircleCI) or test running platform (ex: Sauce Labs, Browser Stack) that will let you run those scripts directly in the environment they provide. Ghost Inspector is really more ideal when managing your tests directly on our service

Let me know if we can help with anything else moving forward.

--
Renata from Ghost Inspector



Click
Make this step optional. (Continue on failure)
Add notes...

#2
.dropdown.active > a[href="#!"].dropdown-toggle
Element text contains ANATOMY

#3
.active > a[href="/en/e-Anatomy"]
Element text contains e-Anatomy

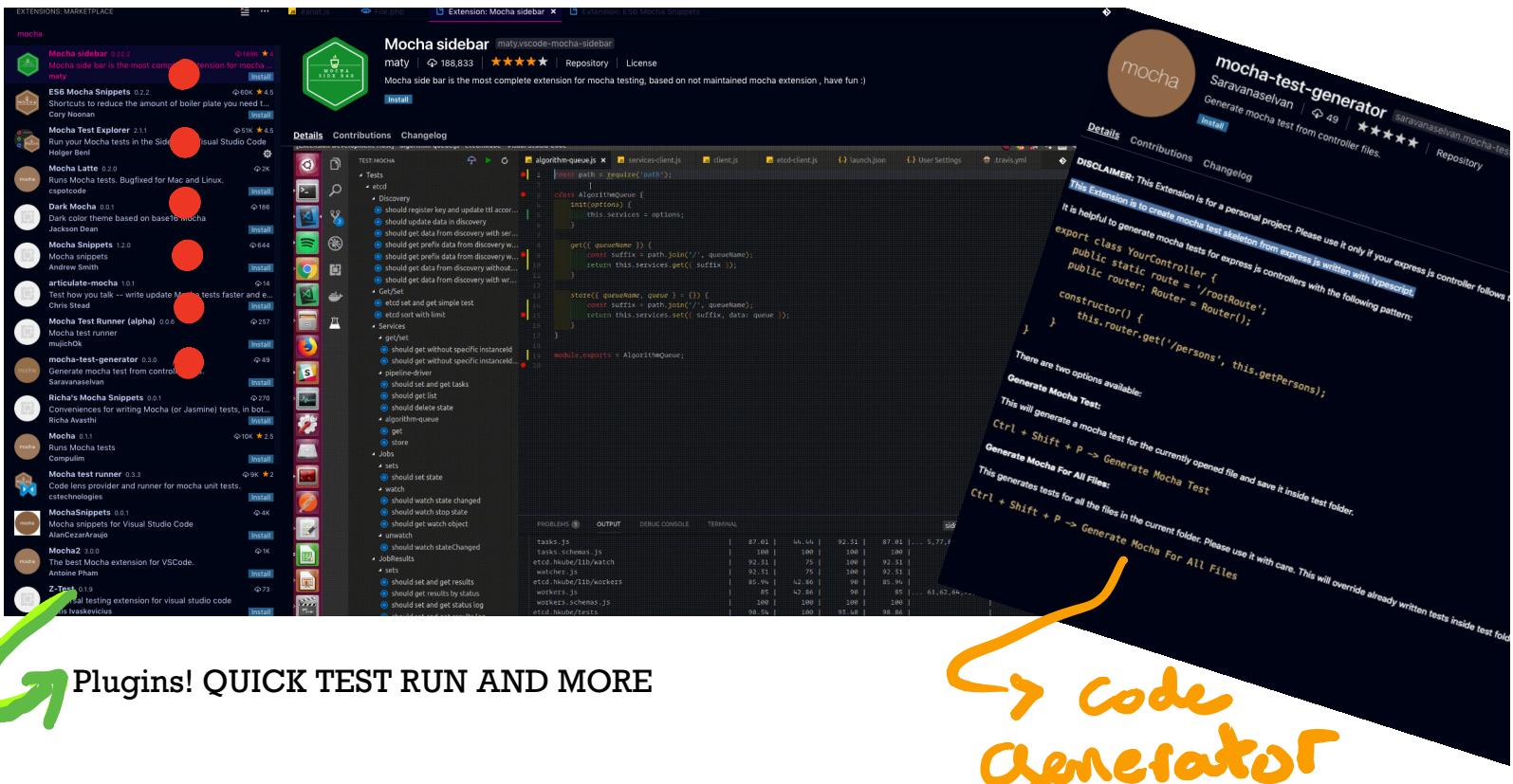
#4
.active > a[href="/en/e-Anatomy"]
Click

#5
Pause (time in ms)
500

#6
a[href="/en/e-Anatomy/Head-and-Neck/Brain-MRI-in-axial-slices"] > .col-xs-5 > img.img-responsive.maximize-width.lazyloaded
Element is present
Make this step optional. (Continue on failure)
Add notes...

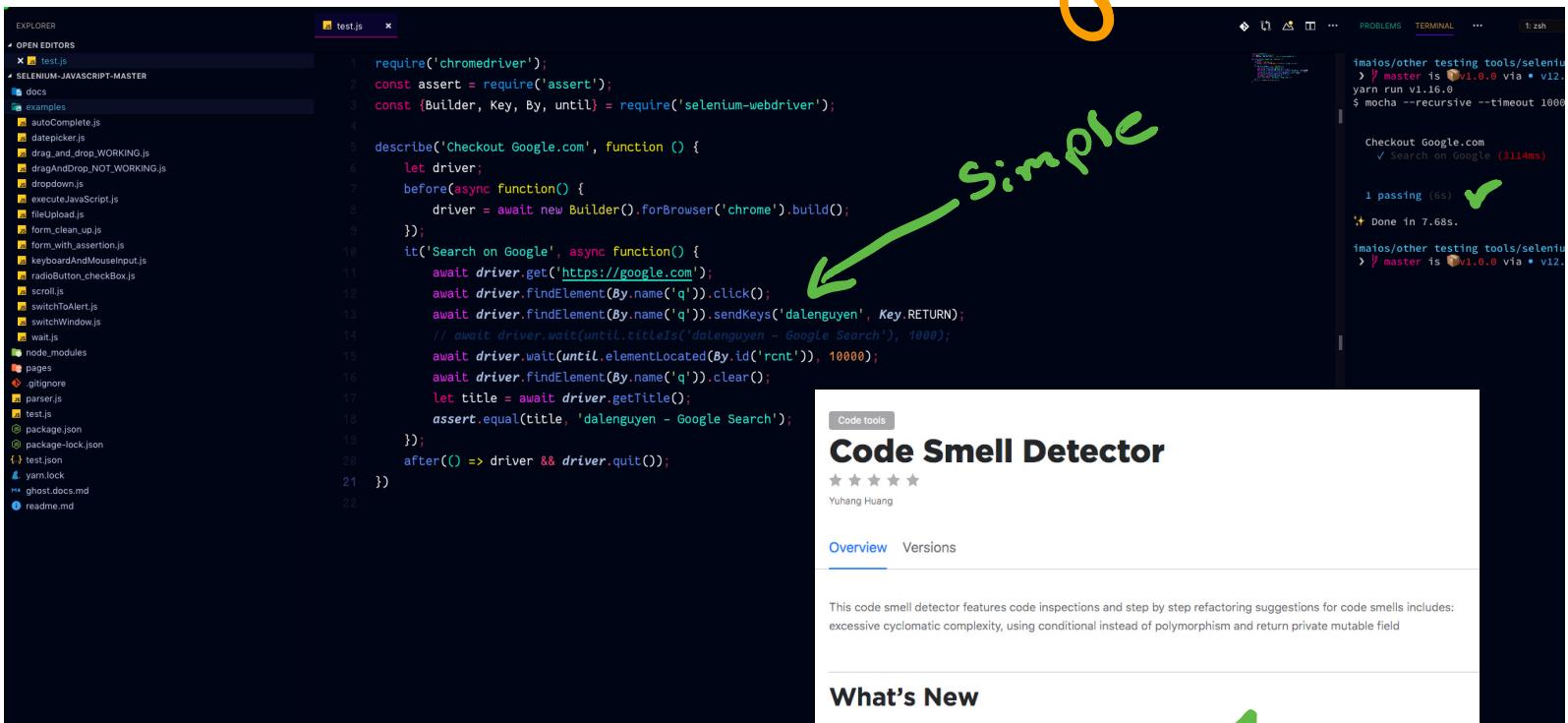
#7
a[href="/en/e-Anatomy/Head-and-Neck/Brain-MRI-in-axial-slices"] > .col-xs-5 > img.img-responsive.maximize-width.lazyloaded
Click
Make this step optional. (Continue on failure)
Add notes...

MORE ON IDE TESTING:



Plugins! QUICK TEST RUN AND MORE

→ code generator



Analyzing Duplicates

PhpStorm helps you find repetitive blocks of code in a certain range. This range can be a single file, a project, a module, or a custom scope. Results of analysis display in the dedicated tab of the [Duplicates tool window](#).

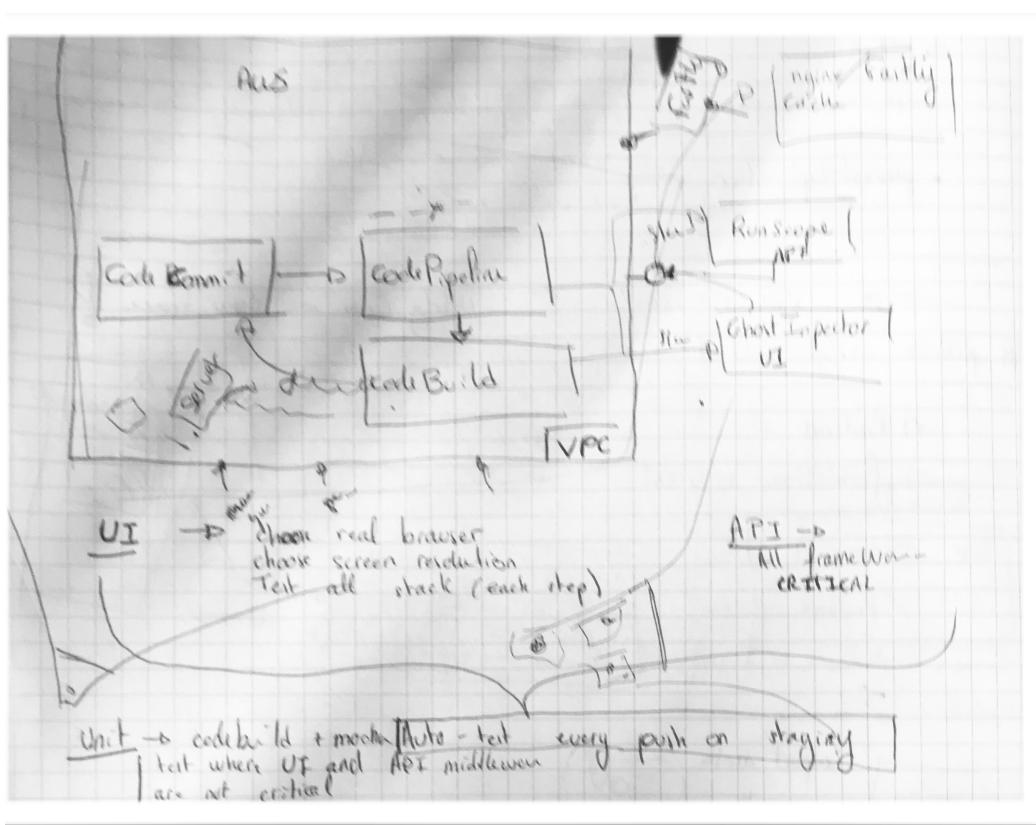
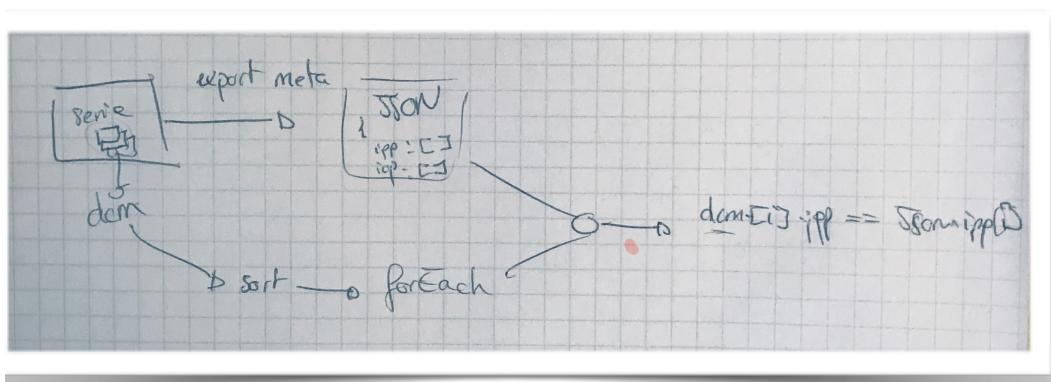
Search for duplicates

- From the main menu, select **Code | Locate Duplicates**.
- In the **Specify Code Duplication Analysis Scope** dialog, specify the analysis scope: whole project, current file, uncommitted files (for the projects under version control), or some custom scope. In addition, you can include test sources into the analysis, too.
- In the **Code Duplication Analysis Settings** dialog, select the language that you want to analyze.

For each language, check the options to define your preferences for the analysis. For example, you can opt to request identical match for code fragments that are considered duplicates, or specify a certain limit below which code constructs are not considered duplicates (to ignore minor variations).

A hand-drawn green arrow points upwards from the bottom left towards the text "The Storm". The text is written in a green, cursive, handwritten-style font.

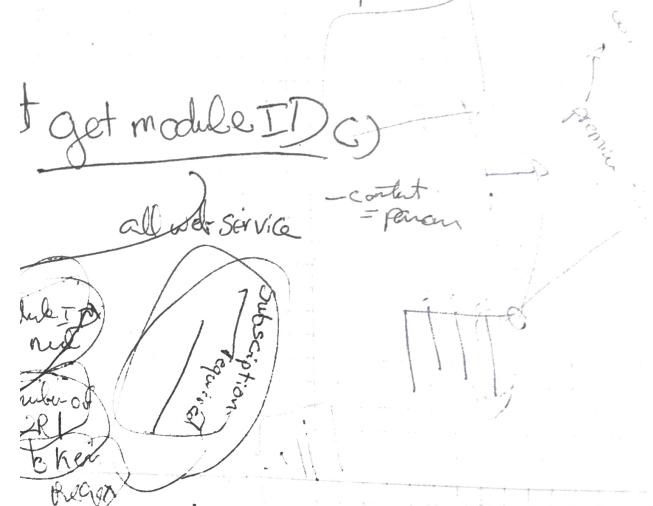
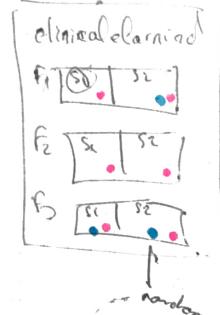
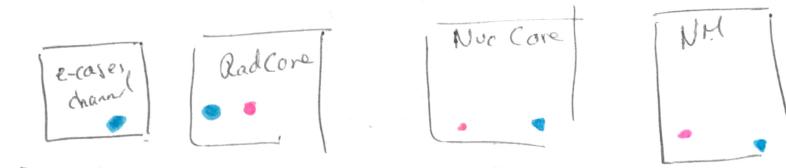
BUT! WHAT WORKS FOR US?



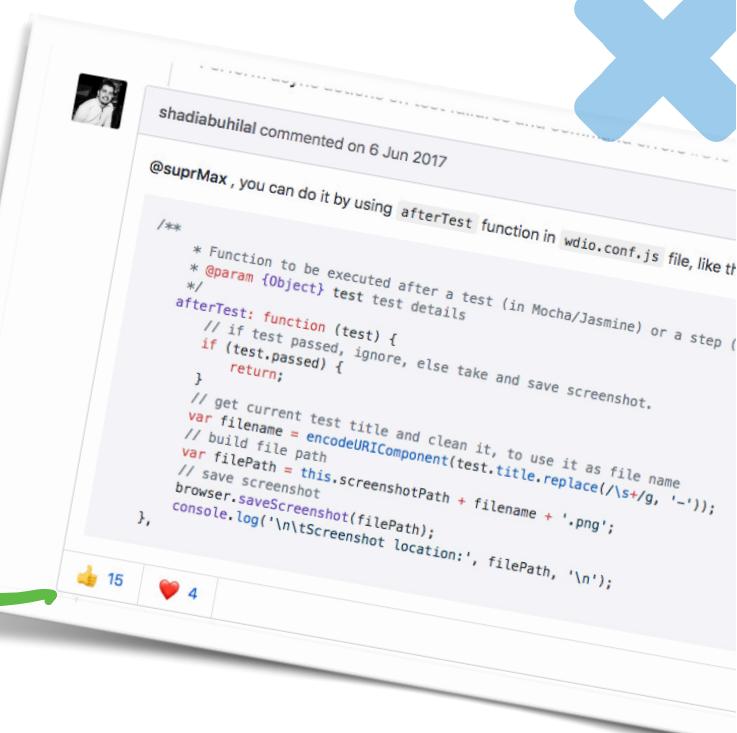
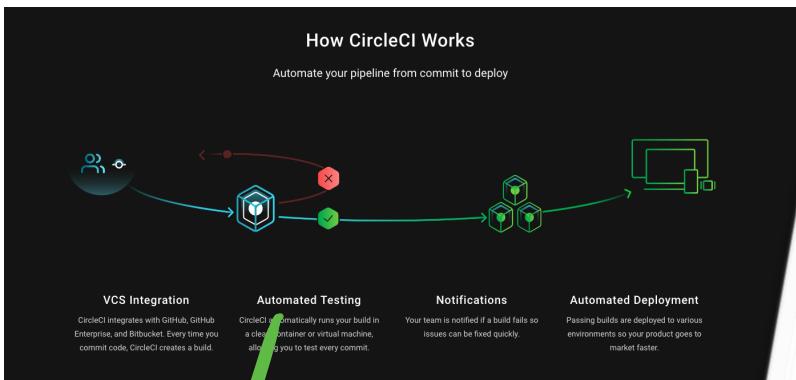
Register AWS, PHP
canal.js

case
get
data [unclear]

Class EASV1.php



Research: External Testing Management



Circle CI
= AWS X
screen shot
handling



The landing page highlights several key features:

- INTEGRATIONS**: **Works with your toolchain**. CircleCI connects with services like GitHub, Bitbucket, Fastlane, Azure, and Slack.
- Everything great teams need**
- Workflows for job orchestration**: Define and orchestrate job execution across multiple supported configurations.
- Choose the CPU/RAM you need**: Easily configure resources to best fit your team's needs.
- Powerful caching**: Speed up builds with expanded caching options.
- Unmatched security**: Includes LDAP for user management, audit logging, and full-level virtual machine isolation.
- First-class Docker support**: Run Docker images, access Docker layer caching, and more.
- Language-agnostic support**: Supports C++, Javascript, .NET, PHP, Python, and Ruby.
- SSH or local builds for easy debugging**: Quickly find and solve problems using SSH access or run jobs in your local environment.
- Insights**: An interactive visual dashboard for monitoring builds.

RUNSCOPE & MOCHA

Runscope

This service provides a simple to use editor to create API tests with no effort.

Pros

- Nice postman-like editor to create API tests.
- Lots of configurable options, initial variables, shared settings, conditionals, etc.
- Uses Chai.js as the Assertion library for responses
- Allows custom scripts to assert responses
- Test scheduling allows you to run tests every X minutes
- Integration with Slack, Jenkins, GitHub, etc.
- Runscope API and trigger URLs allows test customizations
- Can execute tests using multiple regions
- Traffic Inspector shows performance, usage, and error metrics.

Api / UI

WALLABY.JS

Cons

- Custom scripts are not versioned. If your API changes you must change your scripts manually on Runscope.
- Complex asserts must be done in the Custom scripts section.
- Limited NPM packages available.
- No payload or JSON schema validation. Asserts must be specified individually for each property.

Node.js + Mocha Tests

Create all tests using Mocha or other test framework to perform integration tests against the API.

Pros

- Code flexibility. Everything can be done.
- Easy payload and JSON schema validation using libraries.
- Tests can be versioned in GitHub.
- Common triggers exist in Jenkins to execute tests.
- Any integration (e.g. Slack) can be built in code or by including external packages.
- We can reuse these tests for stress testing.

Cons

- Creating tests is considerably more expensive (time).
- Metrics, monitoring, reporting, and integrations must be built manually.
- Scheduling must be configured in Jenkins.
- No multi-region testing.
- Depends on the CI server (not agnostic/external)

Conclusions

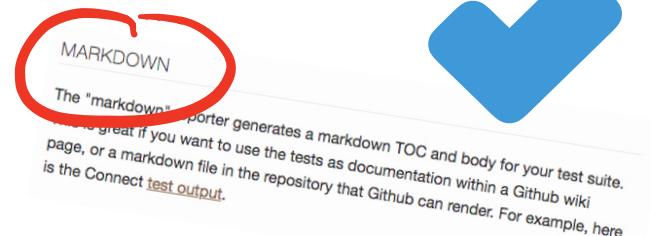
Both approaches have their advantages. Runscope is great for monitoring, metrics, and integrations. Tests are easily created in Runscope. It's a great service to detect errors as soon as possible when monitoring the APIs (e.g. downtime, performance issues, expected HTTP codes, basic asserts).

For business logic, complex asserts, payload validations, custom code is the winner. We have the flexibility to create any type of test and detect errors in the logic of the API. The downside is that we need to code every integration we would like to have, like monitoring, response times, send messages to Slack, etc.

I believe that Runscope is great to start with. We'll have an error detection mechanism that will alert of any possible issue with the APIs providing some basic payload validations. In the meantime, integration tests can be created in parallel to provide a broader coverage.

In the long term, code tests might replace Runscope capabilities and we could keep our automated tests with a proper monitoring infrastructure.

It would be nice if Runscope supported extensibility the way Auth0 supports Rules i.e. in the same way you can write nodejs that executes on Auth0 servers in the authenitcaiton pipeline, it would be great if Runscope could allow devs to execute full fledge Node.js code for complex asserts, using webtask.io.

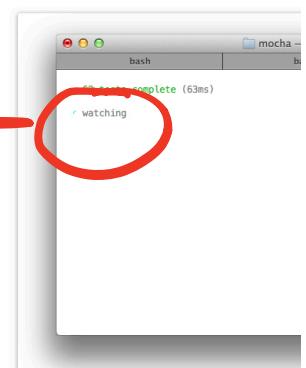


MARKDOWN
The "markdown" reporter generates a markdown TOC and body for your test suite. This is great if you want to use the tests as documentation within a Github wiki page, or a markdown file in the repository that Github can render. For example, here is the Connect [test output](#).

MIN

The "min" reporter displays the summary only, while still outputting to the terminal in order to keep your test summary at the top of the terminal window.

wait



- selenium-html-js (converter html-js)
- examples of files (file examples for selenium)

differences between Selenium conversion formats

first we need to talk about output

HTML : table based structure with td and t(i)s JSON : a series of arrays based on your given tests - easily accessible with. mechanism .SIDE extensions : same thing as JSON but with plugin properties that can be injected straight to the selenium browser extension

JSON : very portable and can be converted with many different libraries into tables including:

1. HTML : table element tags
2. MARKDOWN : table mechanism including ex; |----|
3. JAVASCRIPT : with NPM conversion: <https://www.npmjs.com/package/selenium-html-js-converter>

Ghost Inspector abilities:

pro: able to export a variety of file formats for external tests such as > HTML, JSON AND .SIDE
 con: import can only accept html files.
 > NO PARSERS AVAILABLE TO REIMPORT FOR IDE LOCAL CODING TEST CASES

CONCLUSION : JSON would be the most appropriate and most SUPPORTED mechanism of MANAGING the test results from the overall output of GHOST INSPECTOR

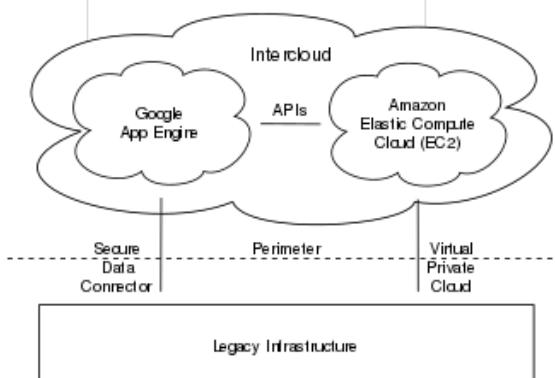
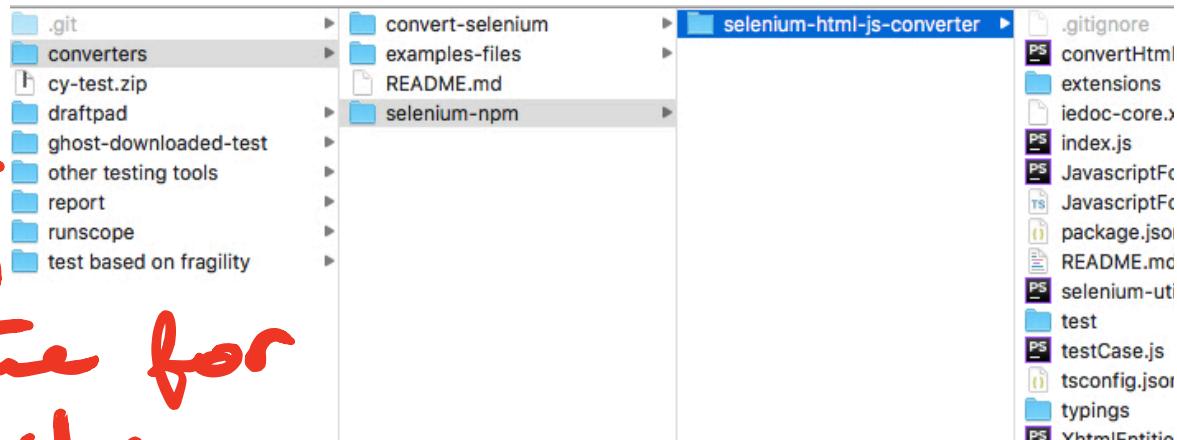
IMPORTANT TOOLS

Extensions : Selenium

1. main plugin that exports different formats : <https://chrome.google.com/webstore/detail/selenium-ide/mooikfahbdckldjjndioackbalphokd>
 2. plugin to obtain certain objects within the page for selenium : <https://chrome.google.com/webstore/detail/selenium-object-finder/afckeoeojpomagchfimplhodjooafmcf>
 3. HTML-JS NPM conversion : <https://www.npmjs.com/package/selenium-html-js-converter>

To do:
 Build json &
 .Side Based
 file structure for
 Portability

2, upload
 to Ghost



Ghost Inspector Node.js Bindings

PASSED

Node.js module for interacting with Ghost Inspector's API.

Installing with npm

```
npm install ghost-inspector
```

Usage

Every method is accessed via your `GhostInspector` instance. Your API Key is passed in when the instance is created:

```
const GhostInspector = require('ghost-inspector')(['api-key']);
```

getSuites([callback])

Fetch an array of all the suites in your account.

```
GhostInspector.getSuites(function (err, suites) {
  if (err) return console.log('Error: ' + err);
  console.log(suites);
});
```

getSuite(suiteId, [callback])

```
bash
-i -X PUT httpbin.org/put -H Content-Type
ation/json -d '{"hello": "world"}'
1 200 OK
ion: keep-alive
unicorn/19.9.0
ti, 02 ● ● ● bash
-Type: $ http PUT httpbin.org/put hello=world
-Length: HTTP/1.1 200 OK
Contro: Access-Control-Allow-Credentials: true
Contro: Access-Control-Allow-Origin: *
1 vegu: Connection: keep-alive
Content-Length: 452
Content-Type: application/json
Date: Fri, 02 Nov 2018 16:53:05 GMT
Server: unicorn/19.9.0
Via: 1.1 vegur
': {}
ers": {}
cept": {}
nnecti: {}
ntent-: {}
ntent-: {}
st": {}
er-Age: {}
": {}
llo": {}
in": {}
": {}
http: {}
```

```
args": {},
"data": "{\"hello\": \"world\"}",
"files": {},
"form": {},
"headers": {
  "Accept": "application/json, */*",
  "Accept-Encoding": "gzip, deflate",
  "Connection": "close",
  "Content-Length": "18",
  "Content-Type": "application/json",
  "Host": "httpbin.org",
  "User-Agent": "HTTPPie/1.0.0"
},
"json": {
  "hello": "world"
},
"origin": "89.102.136.126",
"url": "http://httpbin.org/put"
}
```

install

```
> npm i ghost-inspector
```

weekly downloads

320



version

3.0.1

license

MIT

open issues

0

pull requests

0

homepage

github.com

repository

github

last publish

4 months ago

collaborators



[Test with RunKit](#)

[Report a vulnerability](#)

EAnatomy

parts

stage-greg

tests-not-enabled

DEV-18

15 tests — No Schedule

Staging Clincalearning

0 tests — No Schedule

Check we are testing in staging

Staging test suite

20 tests — No Schedule

Check we are testing in staging

Create New Suite

Folder

(None)

Suite Name

ex: eCommerce Website Tests

Description

ex: A group of tests to check the fun

Create Suite

Changelog

