

Choosing a Web Automation Tool

Sher Singh Bhachu
SinfoniQA

Agenda

- Why JavaScript..?...or Not.
- JavaScript as a tool for QA
- Current tools available + Comparison
- Key things to consider
- Conclusion

Why JavaScript?

- Moving towards FE and Test code living together
- It's not going to die anytime soon
- Becoming 'common' (at least amongst Developers)
- Large community

...or Not

- Fast moving
- Massive transition from Non OO to OO
- Steep learning curve
- Tooling constantly changing, many are left to die
- Sync/Async...callbacks
- Not typically a 'QA' tool
- You're dealing with the JS eco-space (Node, npm, weird ways of doing stuff)

JavaScript as a tool for QA

- Typically 'new' for most
- Not generally used in build/deployment/ops/qa/back-end*
- Pretty much exclusively for Front End development
- Libraries generally cause breaking changes...if they are around for more than a year.

Current tools available

- Webdriver.IO
- Nightwatch.js
- Protractor (used with Angular)
- CasperJS
- Chimp
- WebdriverCSS
- Plugins/Additional supporting tooling
 - Jasmine
 - Mocha/Chai
 - CucumberJS

Common for all tools

- Sit on top of Webdriver
- Page Objects
- Cloud support
- Some form of reporting (3rd party ones can be added)
- Some form of test runner (most are very flexible)
- Support expectation libraries
- Support most browsers with little/no extra work
- Support for Async/Sync

Chart Overview

Tool	Capability		
NightwatchJS	Language chaining, hooks		Node bindings for WebDriver 2, Unit testing
Chimp	Probably best support for Cucumber and Reports	Simian (Reporting/ Documentation)	Wrapper over WebdriverIO, Commercial backing
Protractor	Built in support for Angular	Parallel execution	E2E Testing for Angular
WebdriverIO	Cucumber, Mocha, Jasmine, Mobile support (Appium), Cloud	Applitools (screen compare), Built in common hooks for things like Retry, Very powerful config	Node bindings for WebDriver 2, Wrapper over raw Selenium

Key things to consider...

- Support for Cucumber / Jasmine
- Expectation library support
- Reporting support (HTML or other)
- Async/Sync and ES5/ES6 support (Talk to your local JS developer for more information)

Personal Findings

- What things are most important?
 - Ease of use? Documentation? Support? Roadmap?
- For me...
 - Documentation / Stability
 - Including documenting known 'gotcha's'
 - Flexibility of config and integration with 'typical' front end testing tools (Cucumber, HTML reports)

Useful links/resources

- ES6; <http://es6-features.org/#Constants>
- Gitter; <https://gitter.im>
- Async vs Sync; <http://www.hongkiat.com/blog/synchronous-asynchronous-javascript/>
- Simian; <http://www.simian.io/>