



Nightwatch.js:

An automated way of testing the Primo front-end

Rachelle Anne Orodio- Williams

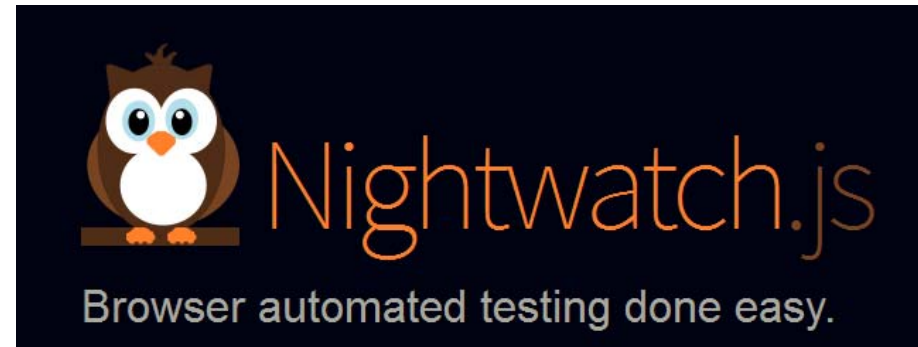
Application Administrator

Monash University Library



Outline

- Why Nightwatch
- Setting it up
- Examples
- What's next?



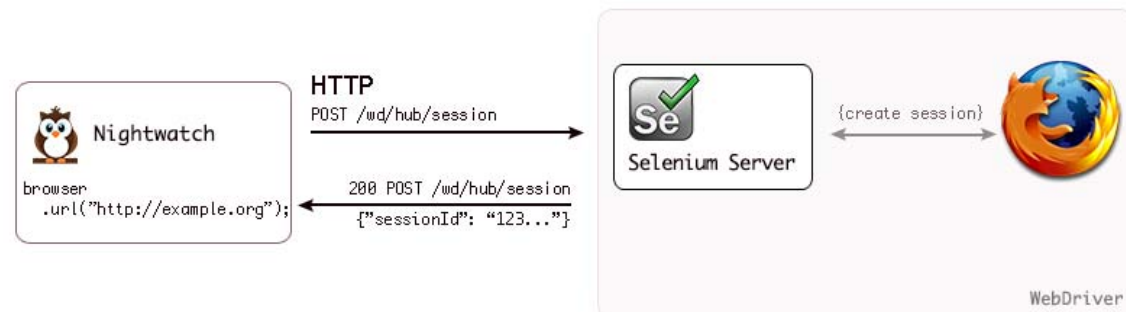
Why Nightwatch?

- Wanted some automated and standard way to run functional testing of the front end
- Open source, free of cost, good GitHub ratings and actively developed
- Learned from IGELU (Uni of Bozen) in September
- SP Upgrade scheduled in November and Platform upgrade in January
- Didn't have time to evaluate other testing applications
- Start getting familiar with the node.js/angular.js framework which is to be used in the new Primo Front end to be released in Aug 2016

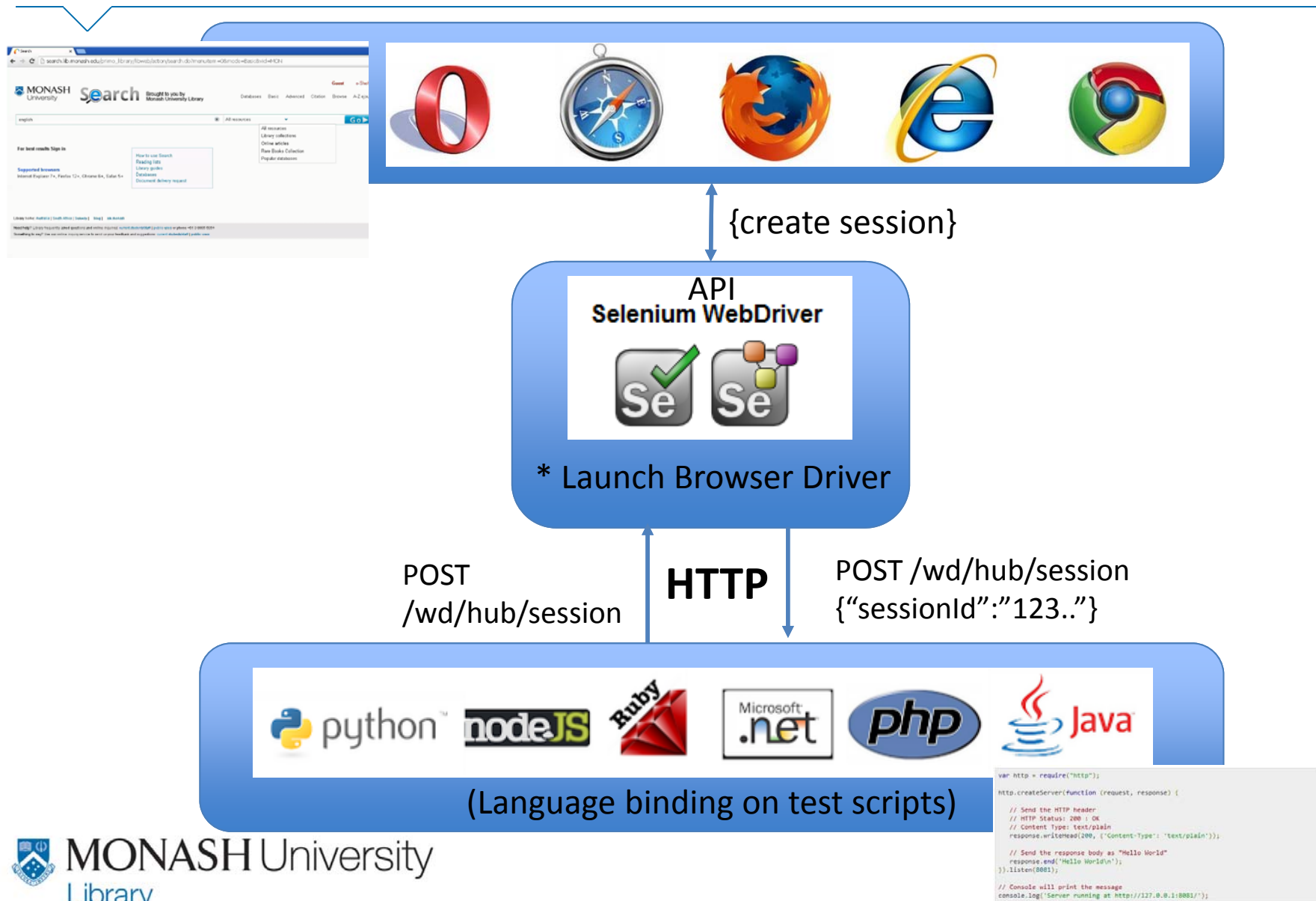


Setting up

- Available for Windows, Linux and MAC
- Encountered issues with ChromeDriver libraries on Linux, so ended up reverting to Windows VM.
- What you need:
 - **Browser** (Chrome, Firefox, IE, etc.)
 - **Selenium 2.0** (Java Servlet that automates browser testing, developed by Google)
 - **Node.js** (platform built on Chrome's JavaScript runtime. Uses event-driven, non-blocking I/O model.)
 - **Nightwatch.js** (automated testing framework written in node.js)



Architecture



Links

■ Nightwatch:

– Windows:

<http://tanmaysarkar.com/nightwatch-setup-in-windows/>

– Linux:

<http://nightwatchjs.org/guide#installation>

■ Selenium: <http://www.seleniumhq.org/>

■ Node.js: <https://nodejs.org/en/>



Example Test Script

```
basic_search_libres - Notepad
File Edit Format View Help
/*
 * Date: 11/11/15
 * Name: basic_search_libres.js
 * Purpose: library collections search
 */
module.exports =
{
  'Primo Basic Search - Library collections' : function (browser)
  {
    browser
      .useCss()
      .windowMaximize()

    /* open Primo (basic search) */
    .url('http://search.lib.monash.edu/primo_library/libweb/action/search.do?menuitem=0&mode=Basic&vid=MON')

    /* wait for the page to load */
    .waitForElementVisible('body', 1000)

    /* search for 'english' */
    .setValue('input[type=text]', 'english')

    /* select library collections then run the search */
    .click('a[id=showMoreOptions]')
    .click('label[for=exlidynamicSelectBodyRadio1]')
    .click('input[id=goButton]')

    /* wait for the results to load */
    .waitForElementVisible('div[id=resultsNumberStile]', 1000)

    /* report if no results found */
    .assert.elementNotPresent('div[class=EXLNoResultsDivTitle]')

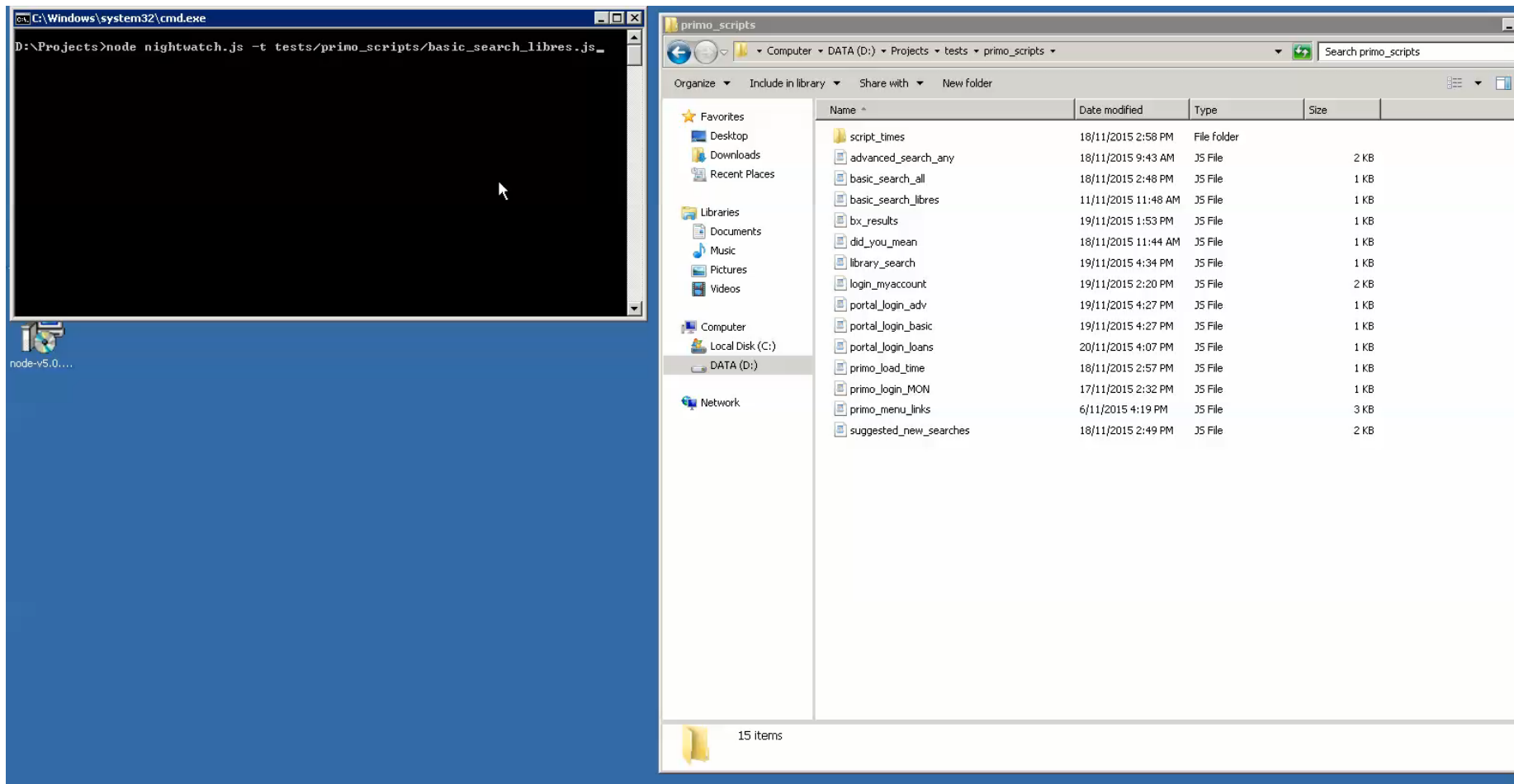
    .end();
  }
};
```

- Open the browser
- Set values to variables
- Check if elements exists
- Click on buttons

Example - Running a test script

■ Basic search within Library Collections

■ `C:\ node nightwatch.js -t tests/primo_scripts/basic_search_libres.js`



The screenshot displays a Windows desktop environment. On the left, a terminal window titled "C:\Windows\system32\cmd.exe" shows the command `D:\Projects>node nightwatch.js -t tests/primo_scripts/basic_search_libres.js` being executed. The terminal output is currently blank. On the right, a File Explorer window titled "primo_scripts" is open, showing the contents of the `D:\Projects\tests\primo_scripts` directory. The window contains a table of files and folders.

Name	Date modified	Type	Size
script_times	18/11/2015 2:58 PM	File folder	
advanced_search_any	18/11/2015 9:43 AM	J5 File	2 KB
basic_search_all	18/11/2015 2:48 PM	J5 File	1 KB
basic_search_libres	11/11/2015 11:48 AM	J5 File	1 KB
bx_results	19/11/2015 1:53 PM	J5 File	1 KB
did_you_mean	18/11/2015 11:44 AM	J5 File	1 KB
library_search	19/11/2015 4:34 PM	J5 File	1 KB
login_myaccount	19/11/2015 2:20 PM	J5 File	2 KB
portal_login_adv	19/11/2015 4:27 PM	J5 File	1 KB
portal_login_basic	19/11/2015 4:27 PM	J5 File	1 KB
portal_login_loans	20/11/2015 4:07 PM	J5 File	1 KB
primo_load_time	18/11/2015 2:57 PM	J5 File	1 KB
primo_login_MON	17/11/2015 2:32 PM	J5 File	1 KB
primo_menu_links	6/11/2015 4:19 PM	J5 File	3 KB
suggested_new_searches	18/11/2015 2:49 PM	J5 File	2 KB

Results & Gotchas

Item to test	Script name	Expected Results	Time elapsed (April 2015 release)	Time elapsed (Nov 2015 release)	Time elapsed (Jan 2016 upgrade)	Time elapsed (Jan 2016 upgrade)	Time elapsed (Jan 2016 upgrade)
			19/11/2015	2/12/2015	1st run	2nd run	3rd run
Check the menu links (eg. Basic Search, Advanced Search, Tags, Help).	<i>primo_menu_links.js</i>	Main menu links are resolving successfully.	63s/63634ms	39.313s	50.301s	36.681s	36.588s
Search using Basic Search, use All Resources scope.	<i>basic_search_all.js</i>	Results should appear	19.111s	18.074s	17.06s	16.499s	12.86s
Search using Basic Search, use Library Collection scope.	<i>basic_search_libres.js</i>	Results should appear	14.1s	14.452s	15.286s	13.936s	14.129s
Search using Advanced Search, looping through "Any" drop	<i>advanced_search_any.js</i>	Advanced Searching should	32.028s	34.693s	33.904s	27.994s	26.958s
Search incorrect search term and see if Did you mean appears	<i>did_you_mean.js</i>	Did you mean appears and	13.644s	**did you mean not appear	15.988s	14.523s	12.598s
Logging in successfully	<i>primo_login_MON.js</i>	User is successfully logged	18.435s	25.031s	39.164s	16.551s	18.087s
Test links under suggested new searches and that they are resolving	<i>suggested_new_searches.js</i>	Links under Suggested search results are resolving	18.985s	17.847s	18.259s	16.427s	18.718s
bx Hot Articles window	<i>bx_results.js</i>	bx Hot article window appearing	16.896s	20.369s	20.887s	18.756s	20.831s
MyAccount - login, loans, requests, fines, blocks and update info	<i>login_myaccount.js</i>	Successfully login and navigate within the page	30.453s	33.135s	26.426s	29.845s	29.365s
Embedded search boxes in portal - basic search	<i>portal_login_basic.js</i>	basic embedded search box on portal working	20.429s	27.223s	27.754s	24.184s	22.447s
Embedded search boxes in portal - advanced search	<i>portal_login_adv.js</i>	advanced embedded search box on portal working	18.141s	19.296s	21.508s	21.215s	20.864s
Embedded search boxes - library website	<i>library_search.js</i>	embedded search box on library website working	23.544s	19.485s	19.005s	21.188s	21.285s
Portal loans information	<i>portal_login_loans.js</i>	Loans information are appearing on the my.monash portal	15.929s	14.494s	16.471s	13.912s	12.971s

- Browser times can vary, depending if it's the first run or succeeding runs.
- Trial and error in identifying CSS components (View It/Get It tabs)

What we need to do next

- Improve scripting (ex. central configuration)
- Not all functionality was tested, still trying to figure how to test certain aspects of the front end (ex. Contents of View It and Get It Tabs)
- Directing errors to log files
- Load testing

Thank you

Rachelle.Williams@monash.edu

