



Automation & Integration Tests Using Nightwatch.js

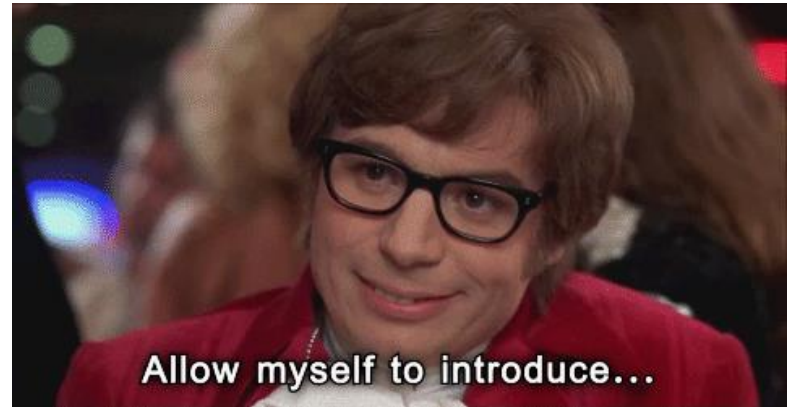
Sitecore User Group Los Angeles

Hugo Santos @coveo

Who am I?

hsantos@coveo.com | @hsantos_x

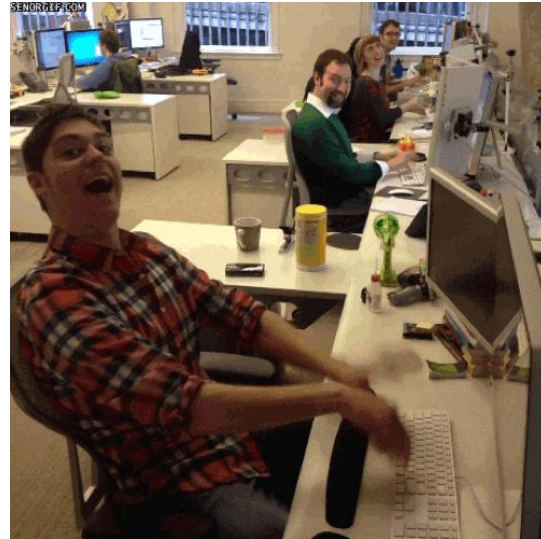
- Hugo Santos
- Software Developer at Coveo since 2016
- 10 years working with .Net
- 5 years working with Sitecore



Why even bother?

Integrate into Sitecore = NotEasyException

- A lot of customization on each implementation
- Coveo supports 3 different major versions



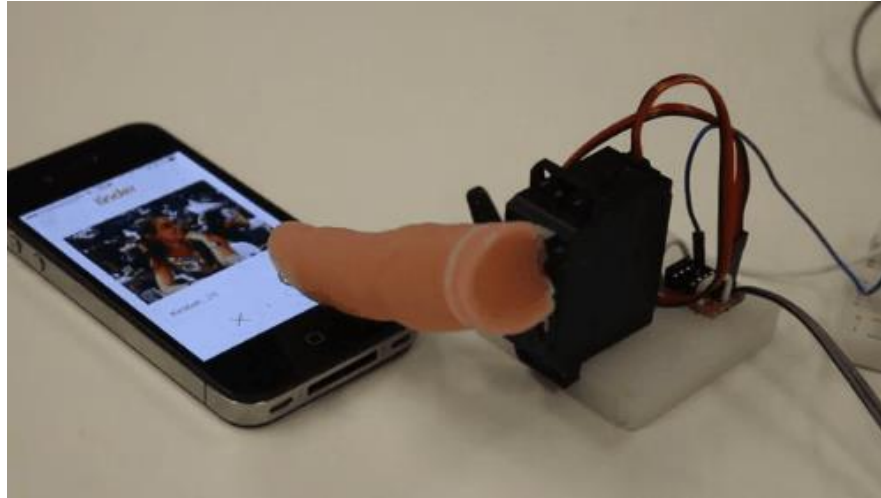
We Want To Release More Often

- Clients don't tend to upgrade once they go live
- 2 months release cycle
- A lot of manual tests during each release cycle



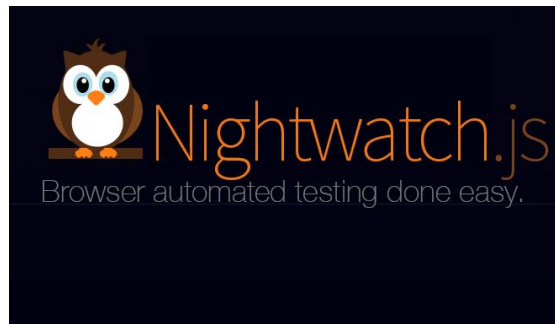
What can we do?

We Need To Automate Everything!



Nightwatch.js

- Browser automated test platform
- Write tests using Javascript or Typescript
- Uses Selenium WebDriver
- Simple syntax
- Other teams at Coveo are already using it



How to use it?

The **Four** Pillars

- Commands
- Assertions
- Pages
- Tests



Nightwatch Commands

- Represents a very specific action
- Nightwatch offers a few basic ones such as:
 - `.pause(2000)`
 - `.click('#nextButton')`
 - `.setValue('input[type=text]', 'Coveo')`
 - `.submit()`
- You can create custom commands

Nightwatch Custom Commands

TS navigateTojQueryModalDialogsFrameFrame.ts ●

```
1 import { NightwatchBrowser, NightwatchCallbackResult } from 'nightwatch';
2 import { globals } from '../globals';
3
4 /**
5  * This custom command can be used to select the frame #jqueryModalDialogsFrame,
6  * that is useful in the Sitecore Control Panel.
7  */
8 export function command(this: NightwatchBrowser): NightwatchBrowser {
9     this.pause(2000);
10    this.frame(null);
11    this.frame('jqueryModalDialogsFrame');
12
13    return this;
14 }
```

Nightwatch Assertions

- Represents a validation
- Nightwatch offers a few basic ones such as:
 - `.assert.title('Coveo Search Interface')`
 - `.assert.containsText('#name', 'Coveo')`
 - `.assert.visible('.coveo-container .coveo-facet a span')`
- You can create custom assertions too

```

-
6  /**
7  * This custom assertion can be used to check if two elements have the same text.
8  */
9  class ElementsHaveSameText implements NightwatchAssertion {
10     public api: NightwatchAPI;
11     public expectedText: string;
12
13     constructor(private firstSelector: string, ...
14     )
15
16     private setExpectedTargetText(value: string) {
17         this.expectedText = value;
18     }
19
20     public expected() {
21         return this.expectedText;
22     }
23
24     public pass(value: string) {
25         return this.expectedText === value;
26     }
27
28     public failure(result: NightwatchCallbackResult) { ...
29     }
30
31     public value(result: NightwatchCallbackResult) {
32         if (result.status === -1) {
33             return '';
34         }
35         return result.value;
36     }
37
38     public command(callback: (this: NightwatchAPI, result: NightwatchCallbackResult) => void) {
39         return this.api.getText(this.firstSelector, callback);
40     }
41 }

```

Nightwatch Pages Object

- Represents a fragment or an entire page
- You must define every Page Object used by your tests
- Can call commands and assertions
- Contains methods that perform major actions
 - `.republishSite()`
 - `.rebuildCoveoIndexes()`
 - `.rebuildLuceneIndexes()`

Nightwatch Pages Object

TS sitecoreContentEditor.ts x

```
5
6  export interface ISitecoreContentEditor {
7    searchHomeItem(this: this): this;
8    republishSite(this: this): this;
9  }
10
11 const sitecoreContentEditor: ISitecoreContentEditor = {
12   searchHomeItem: function (this: INightwatchSitecoreContentEditor): INightwatchSitecoreContentEditor {
13     this.waitForElementVisible('@searchBox', globals.waitForElementTimeout, 'The content editor searchbox is visible')
14     .setValue('@searchBox', 'Home')
15     .click('@searchButton')
16     .waitForElementVisible('@homeSearchResult', globals.waitForElementTimeout, 'The Home item is returned as result.');
```

```
17   return this;
18 },
19
20   republishSite: function (this: INightwatchSitecoreContentEditor): INightwatchSitecoreContentEditor {
21     if (state.currentSitecoreMajorVersion == 7) {
22       this.api.page.sitecore7ContentEditor().republishSite();
23     } else {
24       this.api.page.sitecore8ContentEditor().republishSite();
25     }
26     return this;
27   }
28 };
29
30 export const url = globals.scInstanceURL + globals.scURL.contentEditor;
31 export const commands = [sitecoreContentEditor];
32 export const elements = {
33   searchBox: '#TreeSearch',
34   searchButton: '.scSearchButton',
```

Nightwatch Tests

- Represents a unit of test
- Uses one or more pages
- Can call commands and assertions
- Organized in sub-sections

Nightwatch Tests

TS searchPageFeatures.ts x

```
7  if (!globals.scUserName) { throw new Error('You must define the scUserName parameter'); }
8  if (!globals.scUserPassword) { throw new Error('You must define the scUserPassword parameter'); }
9
10 export = {
11   beforeEach: function (client: NightwatchBrowser): void {
12     client.maximizeWindow();
13   },
14
15   'Login to Sitecore Instance': function (client: NightwatchBrowser): void {
16     client.page.sitecoreLogin()
17       .navigate()
18       .loginSitecore();
19   },
20
21   'Publish from master to web': function (client: NightwatchBrowser): void {
22     client.page.sitecoreContentEditor()
23       .navigate()
24       .republishSite();
25   },
26
27   'Rebuild the Coveo indexes': function (client: NightwatchBrowser): void {
28     client.page.sitecoreControlPanel()
29       .navigate()
30       .rebuildCoveoIndexes();
31   },
32
33   '[CoveoHive] Test if a search using a facet is working': function (client: NightwatchBrowser): void {
34     client.page.coveoSearchPage()
35       .clickOnFacetValue(UiType.Hive)
36       .assert.elementsHaveSameText(elements.searchPageTotalResults, elements.facetTestTemplateOptionCount, 'The search
37   },
38
```

How to Run A Test?

- Define scripts in the package.json for one or more tests
- Call it using [npm run script name](#)
- Pass parameters using environment variables

```
14   "nightwatch": "^0.9.21",
15   "path": "0.12.7",
16   "request": "2.85.0",
17   "selenium-server-standalone-jar": "2.53.1",
18   "typescript": "2.7.2"
19 },
20 "scripts": {
21   "build": "tsc",
22   "watch": "tsc --watch",
23   "searchPageFeatures": "./node_modules/.bin/nightwatch --test bin/tests/searchPageFeatures.js --env local"
24 }
25 }
26 }
```

That's enough,
Let's see some **action!**

The real **Impact**

The Current Status

- Spawn a machine for each Sitecore version
- Automate the installation of our packages
- Automate the configuration of the package
- Automate the republish
- Automate the Coveo index full rebuild
- Test the most basic search features

What's **Next?**

Work in progress...

- More tests for search page features
- Separate **tests** from automation **tasks**
- Open source project?



Questions?