

Join The Darkside

Selenium testing with Nightwatch.js

presented by seth mclaughlin on 10.21.14







Venus.js
where bugs go to die

SH-PE



Nightwatch.js

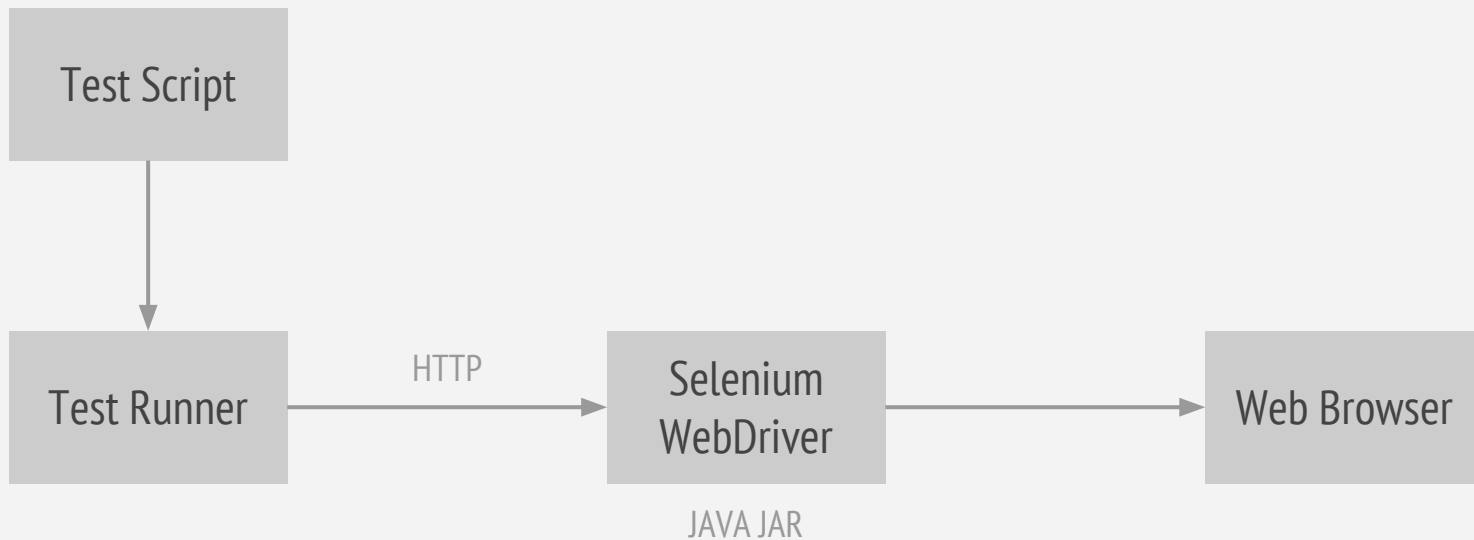


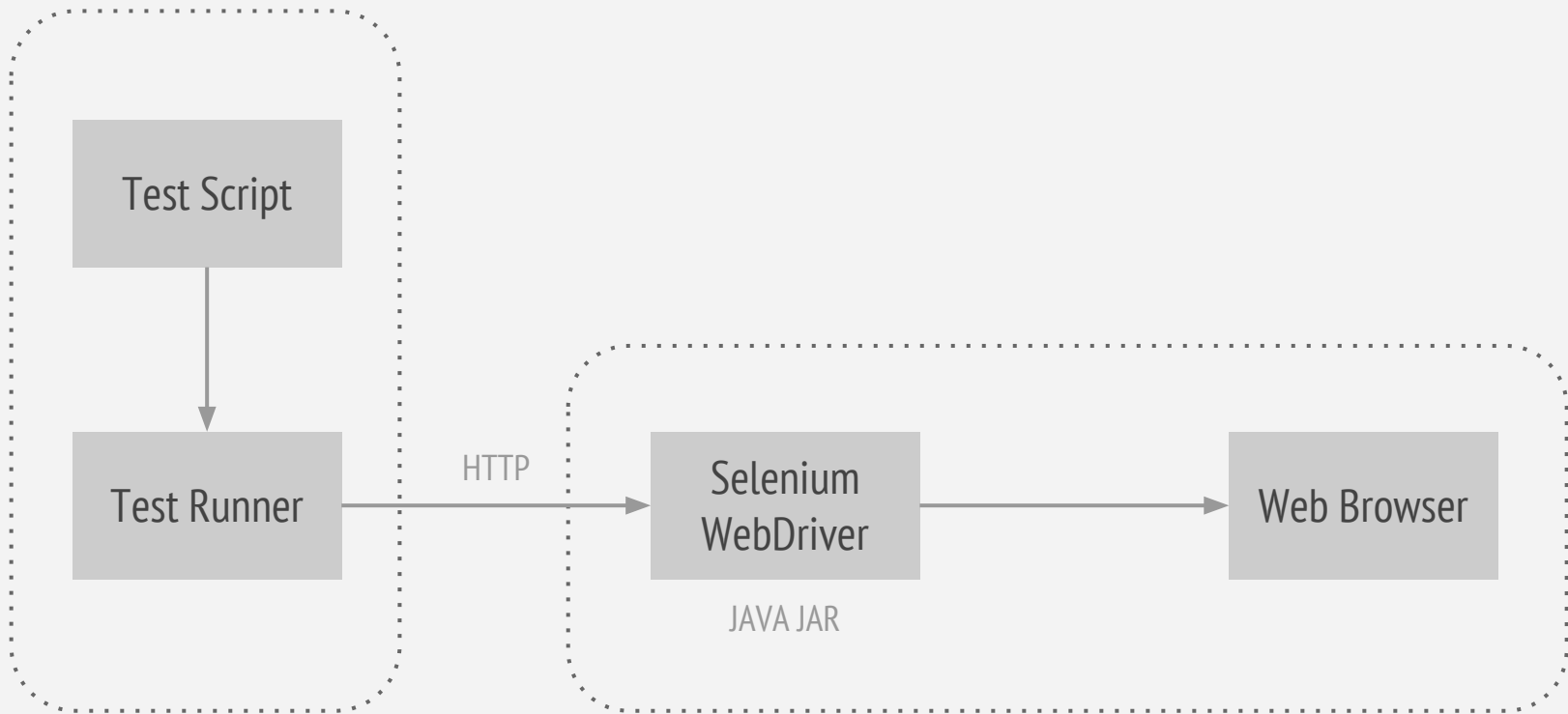
learn more: www.nightwatchjs.org

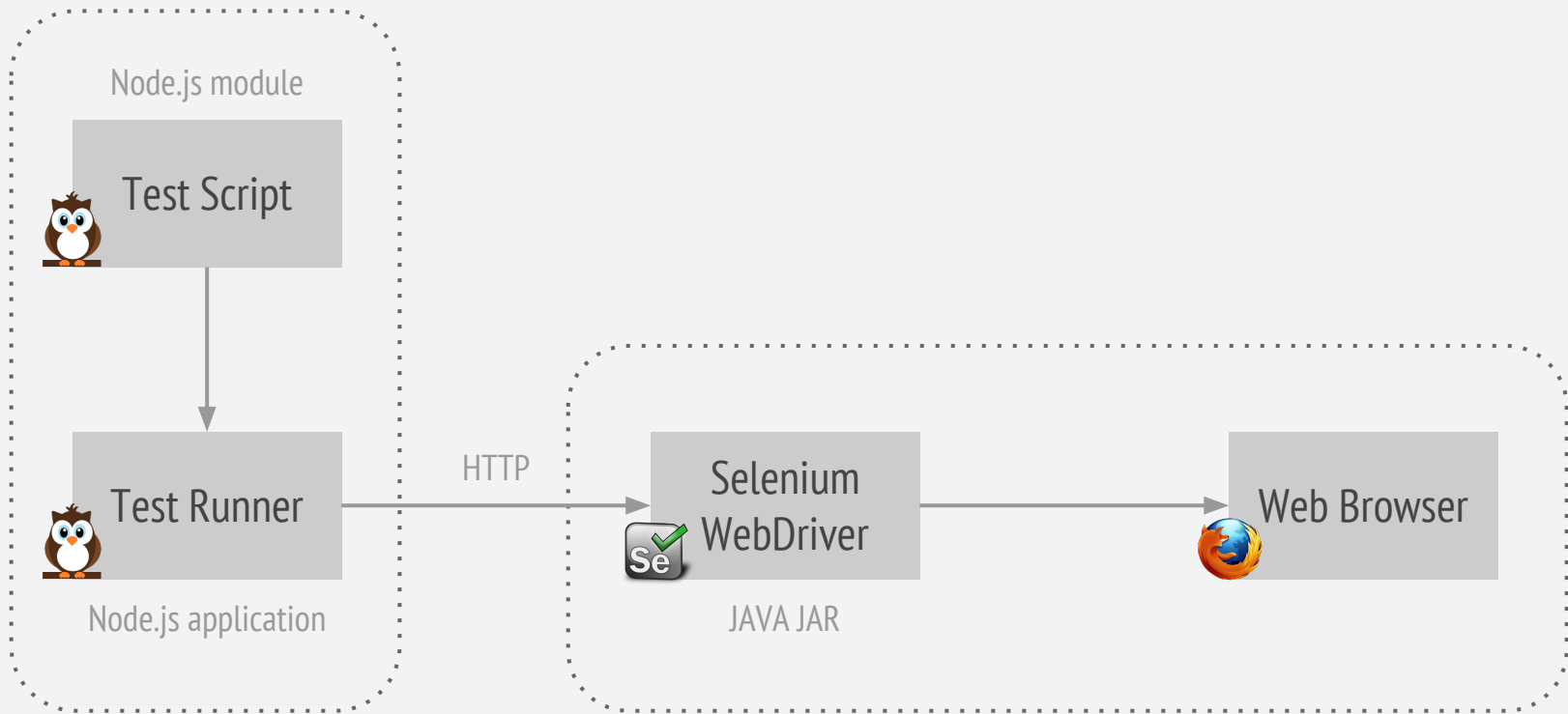
1. Project Overview
2. Features
3. Getting Started

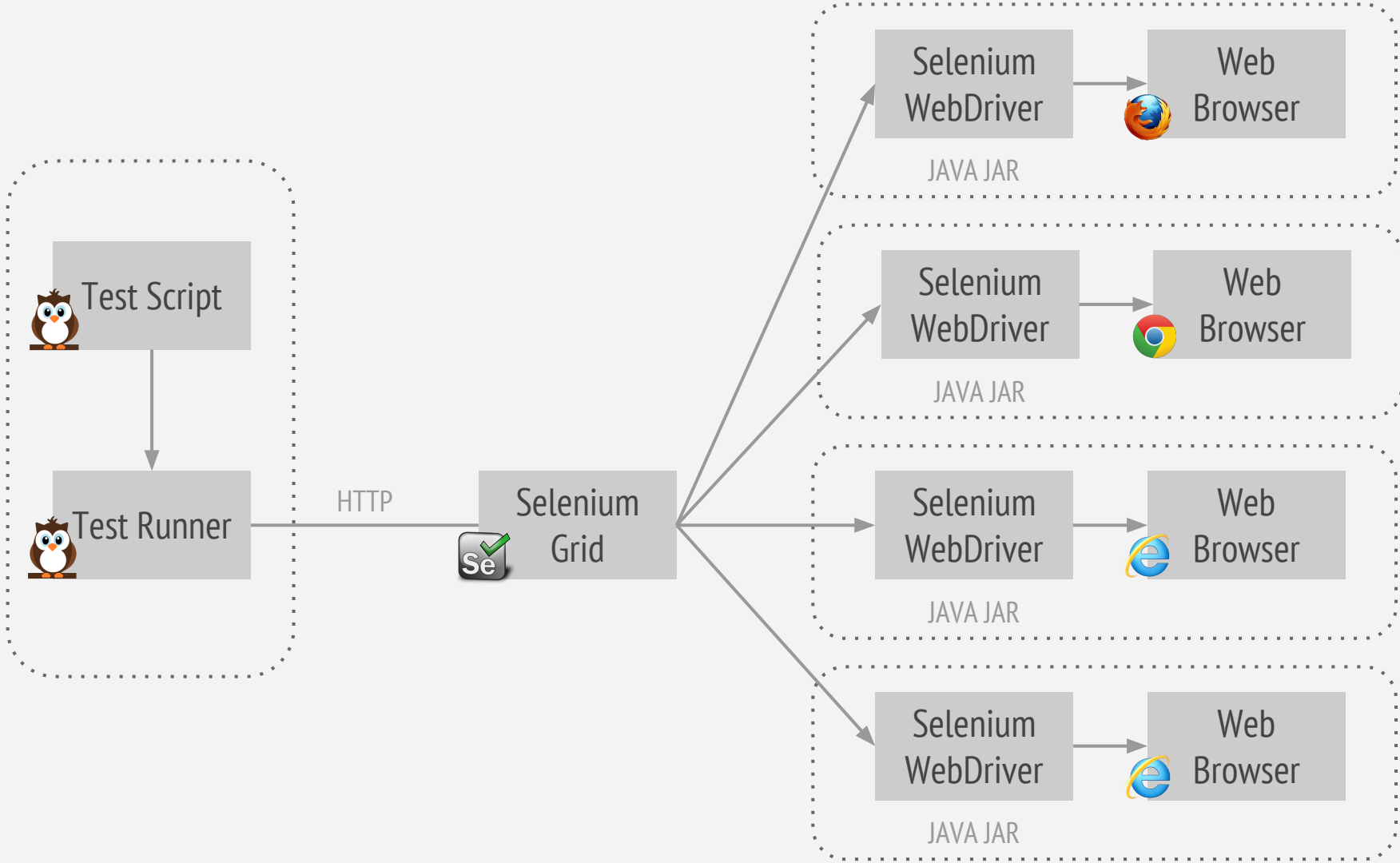
"End to End" testing

1. Navigate to Login page
2. Locate username form field, type in **seth**
3. Locate password form field, type in **html5dev**
4. Locate submit form button and click it
5. Wait for form submission to complete
6. Verify that title of page is now equal to **Welcome!**











Features

- ★ Good documentation
- ★ Use CSS or XPATH selectors
- ★ Test runner can execute sequentially or in parallel
- ★ Test filtering by file name pattern, folders and tags
- ★ SauceLabs + BrowserStack support
- ★ Built in JUnit XML reporting
- ★ Extension model for custom commands

created by
Andrei Rusu



@beatfactor



~9,000
downloads per month

~2,200
stars on github

~150
forks on github

~83%
code coverage



nightwatch



A node.js bindings implementation for selenium
2.0/webdriver

```
$ npm install nightwatch
```

Want to see pretty graphs? [Log in now!](#)

142 downloads in the last day
1 967 downloads in the last week
9 288 downloads in the last month

Last Published By



beatfactor

Version

0.5.31 last updated 5 days ago

learn more: www.npmjs.org/package/nightwatch

27 contributors

1 primary contributor (Andrei)

1 pending pull requests (from 8/14/14)

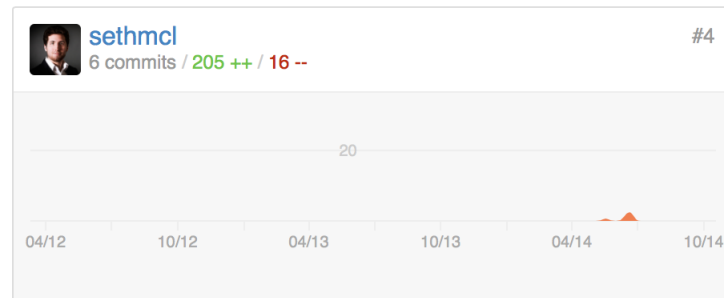
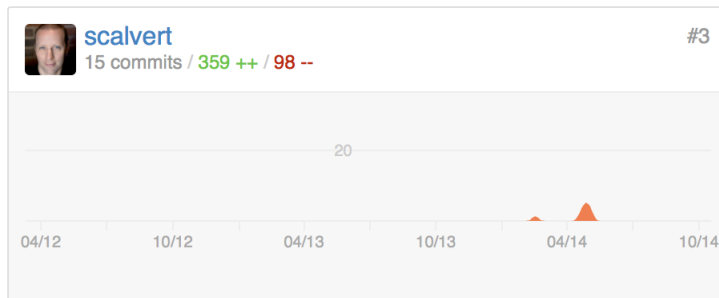
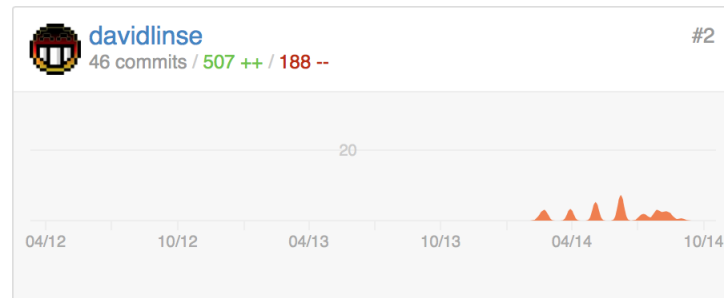
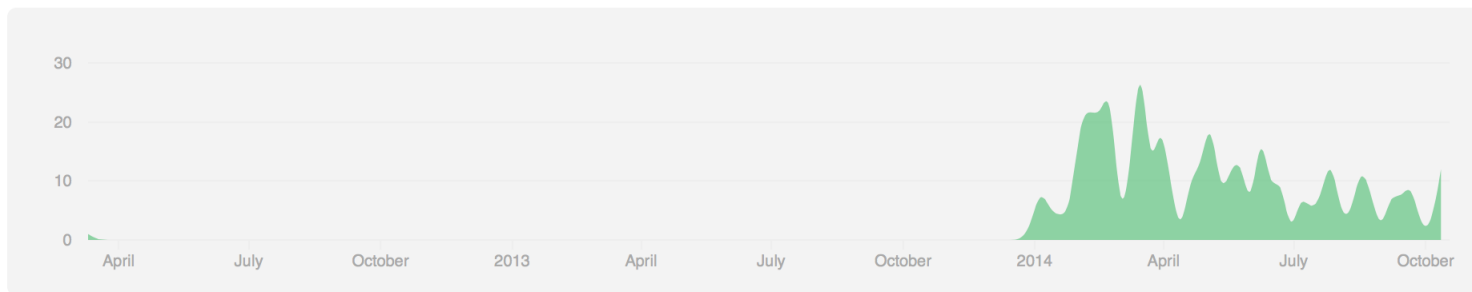
46 open issues

164 closed issues

Mar 11, 2012 – Oct 18, 2014

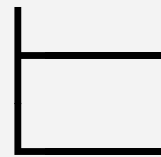
Contributions: **Commits** ▾

Contributions to master, excluding merge commits



Sample test: Login flow

1. Navigate to Login page
2. Locate username form field, type in **seth**
3. Locate password form field, type in **html5dev**
4. Locate submit form button and click it
5. Wait for form submission to complete
6. Verify that title of page is now equal to **Welcome!**



```
Nightwatch.js
tests
└─ login.js
```

```
| Nightwatch.js  
| tests  
| └─ login.js
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```



```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```


```
module.exports = {  
  'Fill out form and login': function (client) {  
    client.url('http://localhost:8000');  
    client.assert.title('Awesome App');  
    client.setValue('input[name=username]', 'seth');  
    client.setValue('input[name=password]', 'html5dev');  
    client.click('input[type=submit]');  
    client.assert.title('Welcome!');  
    client.end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client.url('http://localhost:8000');  
    client.assert.title('Awesome App');  
    client.setValue('input[name=username]', 'seth');  
    client.setValue('input[name=password]', 'html5dev');  
    client.click('input[type=submit]');  
    client.assert.title('Welcome!');  
    console.log('all done!'); // wrong!  
    client.end();  
  }  
};
```




login.js


```
module.exports = {  
  'Fill out form and login': function (client) {  
    client.url('http://localhost:8000');  
    client.assert.title('Awesome App');  
    client.setValue('input[name=username]', 'seth');  
    client.setValue('input[name=password]', 'html5dev');  
    client.click('input[type=submit]');  
    client.assert.title('Welcome!');  
    client.perform(function (client, done) {  
      console.log('all done!'); // right!  
    });  
    client.end();  
  }  
};
```



login.js

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client.url('http://localhost:8000');  
    client.assert.title('Awesome App');  
    client.setValue('input[name=username]', 'seth');  
    client.setValue('input[name=password]', 'html5dev');  
    client.click('input[type=submit]');  
    client.assert.title('Welcome!');  
    client.perform(function (client, done) {  
      foo.doSomethingAsync().then(done);  
    });  
    client.end();  
  }  
};
```





```
├─ Nightwatch.js
├─ tests
│   └─ login.js
```


Nightwatch.js

```
module.exports = {
  src_folders: ['./tests'],
  output_folder: './results',

  selenium: {
    start_process: true,
    server_path: './selenium-server-standalone-2.38.0.jar',
    log_path: './results',
    host: '127.0.0.1',
    port: 4444
  },

  test_settings: {
    default: {
      selenium_host: '127.0.0.1',
      selenium_port: 4444,
      screenshots: {
        enabled: true
      }
    }
  }
}
```

Nightwatch.js

```
module.exports = {
  src_folders: ['./tests'],
  output_folder: './results',

  selenium: {
    start_process: true,
    server_path: './selenium-server-standalone-2.38.0.jar',
    log_path: './results',
    host: '127.0.0.1',
    port: 4444
  },

  test_settings: {
    default: {
      selenium_host: '127.0.0.1',
      selenium_port: 4444,
      screenshots: {
        enabled: true
      }
    }
  }
}
```

Nightwatch.js

```
module.exports = {
  src_folders: ['./tests'],
  output_folder: './results',

  selenium: {
    start_process: true,
    server_path: './selenium-server-standalone-2.38.0.jar',
    log_path: './results',
    host: '127.0.0.1',
    port: 4444
  },

  test_settings: {
    default: {
      selenium_host: '127.0.0.1',
      selenium_port: 4444,
      screenshots: {
        enabled: true
      }
    }
  }
}
```

Nightwatch.js

```
module.exports = {  
  src_folders: ['./tests'],  
  output_folder: './results',  
  
  selenium: {  
    start_process: true,  
    server_path: './selenium-server-standalone-2.38.0.jar',  
    log_path: './results',  
    host: '127.0.0.1',  
    port: 4444  
  },  
  
  test_settings: {  
    default: {  
      selenium_host: '127.0.0.1',  
      selenium_port: 4444,  
      screenshots: {  
        enabled: true  
      }  
    }  
  }  
}
```

Nightwatch.js

```
module.exports = {
  src_folders: ['./tests'],
  output_folder: './results',

  selenium: {
    start_process: true,
    server_path: './selenium-server-standalone-2.38.0.jar',
    log_path: './results',
    host: '127.0.0.1',
    port: 4444
  },

  test_settings: {
    default: {
      selenium_host: '127.0.0.1',
      selenium_port: 4444,
      screenshots: {
        enabled: true
      }
    }
  }
}
```

Nightwatch.js

```
port: 4444
},

test_settings: {
  default: {
    selenium_host: '127.0.0.1',
    selenium_port: 4444,
    screenshots: {
      enabled: true,
      path: './results/screenshots'
    },
    desiredCapabilities: {
      browserName: 'firefox',
      javascriptEnabled: true,
      acceptSslCerts: true
    }
  }
}
};
```

Nightwatch.js

```
port: 4444
},

test_settings: {
  default: {
    selenium_host: '127.0.0.1',
    selenium_port: 4444,
    screenshots: {
      enabled: true,
      path: './results/screenshots'
    },
    desiredCapabilities: {
      browserName: 'firefox',
      javascriptEnabled: true,
      acceptSslCerts: true
    }
  }
}
};
```

Nightwatch.js

```
port: 4444
},

test_settings: {
  default: {
    selenium_host: '127.0.0.1',
    selenium_port: 4444,
    screenshots: {
      enabled: true,
      path: './results/screenshots'
    },
    desiredCapabilities: {
      browserName: 'firefox',
      javascriptEnabled: true,
      acceptSslCerts: true
    }
  }
};
```


Nightwatch.js

```
port: 4444
},

test_settings: {
  default: {
    selenium_host: '127.0.0.1',
    selenium_port: 4444,
    screenshots: {
      enabled: true,
      path: './results/screenshots'
    },
    desiredCapabilities: {
      browserName: 'firefox',
      javascriptEnabled: true,
      acceptSslCerts: true
    }
  }
};
```

```
> nightwatch -c ./Nightwatch.js --env default
```

[Login] Test Suite

=====

Running: Fill out form and login


- ✓ Testing if the page title equals "Awesome App".
- ✓ Testing if the page title equals "Welcome!".

OK. 2 total assertions passed. (2.233s)

```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```



```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

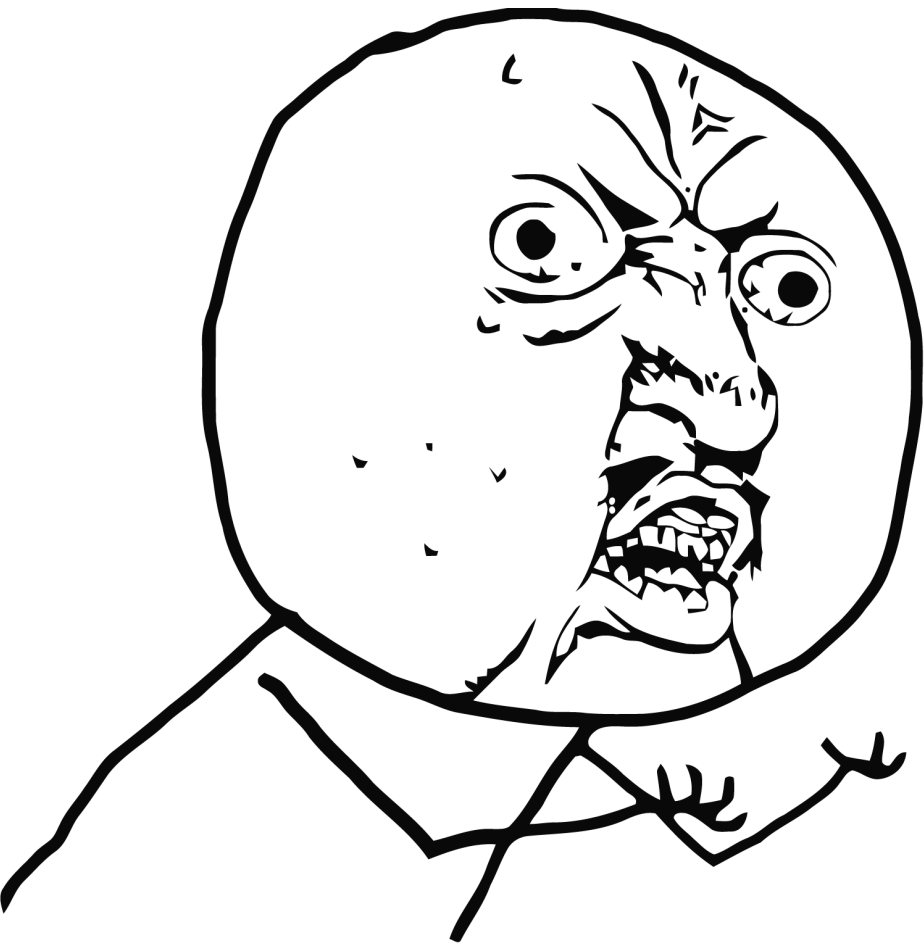


```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```



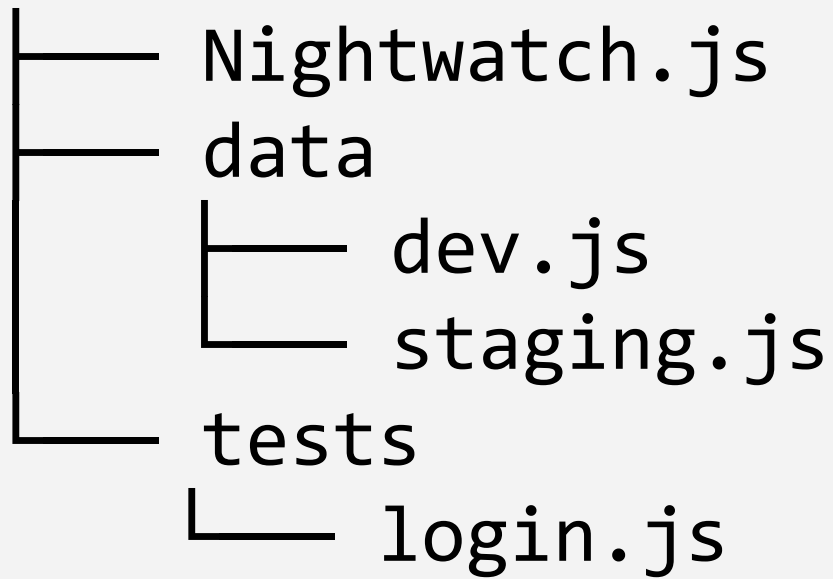
```
module.exports = {  
  'Fill out form and login': function (client) {  
    client  
      .url('http://localhost:8000')  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', 'seth')  
      .setValue('input[name=password]', 'html5dev')  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```





**Y U HARDCODE
VALUES ???**

Data Driven Tests



```
module.exports = {  
  username: 'seth',  
  password: 'html5dev',  
  urls: {  
    login: 'http://localhost:8000'  
  }  
};
```

Nightwatch.js

```
test_settings: {  
  default: {  
    selenium_host: '127.0.0.1',  
    selenium_port: 4444,  
    screenshots: {  
      enabled: true,  
      path: './results/screenshots'  
    },  
    desiredCapabilities: {  
      browserName: 'firefox',  
      javascriptEnabled: true,  
      acceptSslCerts: true  
    },  
    globals: require('./data/dev')  
  }  
}
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    var data = client.globals;  
  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password)  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    var data = client.globals;  
  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password)  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```


```
module.exports = {  
  'Fill out form and login': function (client) {  
    var data = client.globals;  
  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password)  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```

```
module.exports = {  
  'Fill out form and login': function (client) {  
    var data = client.globals;  
  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password)  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  }  
};
```



```
module.exports = {  
  'Fill out form': function (client) {  
    var data = client.globals;  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password);  
  },  
  
  'Submit form': function (client) {  
    var data = client.globals;  
    client  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  },  
};
```

```
module.exports = {  
  'Fill out form': function (client) {  
    var data = client.globals;  
    client  
      .url(data.urls.login)  
      .assert.title('Awesome App')  
      .setValue('input[name=username]', data.username)  
      .setValue('input[name=password]', data.password);  
  },  
  
  'Submit form': function (client) {  
    console.log('previous step is done executing');  
    var data = client.globals;  
    client  
      .click('input[type=submit]')  
      .assert.title('Welcome!')  
      .end();  
  },  
};
```



[Login] Test Suite

=====

Running: **Fill out form**

✓ Testing if the page title equals "Awesome App".

OK. 1 assertions passed. (1.567s)

Running: **Submit form**

✓ Testing if the page title equals "Welcome!".

OK. 1 assertions passed. (204ms)

Terminology

Login

Test suite

Fill out form

Test

Submit form

Test

Assert vs. Verify

```
assert.title('Awesome App')
```

If false, log failure
and **stop** running current
test suite

```
verify.title('Awesome App')
```

If false, log failure
and **continue** running
current test suite

Debugging tips

`pause()` Pause test execution, leaving browser window open.

`client`

```
.url('http://localhost:8000')  
.assert.title('Awesome App')  
.setValue('input[name=username]', 'seth')  
.setValue('input[name=password]', 'html5dev')  
→ .pause()  
.click('input[type=submit]')
```

Debugging tips

debugger; Insert breakpoint (in node.js code)

```
> node debug `which nightwatch`  
    -c ./Nightwatch.js --env default
```

Commands: run (r), cont (c), next (n), step (s), out (o), backtrace (bt), setBreakpoint (sb), clearBreakpoint (cb), watch, unwatch, watchers, repl, restart, kill, list, scripts, breakOnException, breakpoints, version

Commands

clearValue
click
deleteCookie
deleteCookies
end
getAttribute
getCookie
getCookies
getCssProperty
getElementSize
getLocation

getLocationInView
getTagName
getText
getTitle
getValue
init
injectScript
isVisible
maximizeWindow
moveToElement
pause

resizeWindow
saveScreenshot
setCookie
setValue
submitForm
switchWindow
urlHash
waitForElementNotPresent
waitForElementNotVisible
waitForElementPresent
waitForElementVisible

Assertions

attributeEquals

containsText

cssClassPresent

cssClassNotPresent

cssProperty

elementPresent

elementNotPresent

hidden

title

urlContains

value

valueContains

visible

Using custom commands and assertions

```
module.exports = {  
  'Load Netflix.com': function (client) {  
    client  
      .url('http://www.netflix.com')  
      .tagCount('a', function (result) {  
        console.log(  
          'NOTE: there are %s anchor elements on the page\n',  
          result.value  
        );  
      })  
      .assert.tagCountGreaterThan('a', 100)  
      .end();  
  }  
};
```

Using custom commands and assertions

```
module.exports = {  
  'Load Netflix.com': function (client) {  
    client  
      .url('http://www.netflix.com')  
      .tagCount('a', function (result) {  
        console.log(  
          'NOTE: there are %s anchor elements on the page\n',  
          result.value  
        );  
      })  
      .assert.tagCountGreaterThan('a', 100)  
      .end();  
  }  
};
```

Using custom commands and assertions

```
module.exports = {  
  'Load Netflix.com': function (client) {  
    client  
      .url('http://www.netflix.com')  
      .tagCount('a', function (result) {  
        console.log(  
          'NOTE: there are %s anchor elements on the page\n',  
          result.value  
        );  
      })  
      .assert.tagCountGreaterThan('a', 100)  
      .end();  
  }  
};
```

Extending Nightwatch: custom commands

```
// command to return the number of elements in a page
// which are of a certain tag name
exports.command = function (tagName, callback) {
  callback = callback || function () {};

  this.execute(function (tagName) {
    return document.getElementsByTagName(tagName).length;
  }, [tagName], function (result) {
    callback.call(this, result);
  });

  return this; // allows the command to be chained.
};
```

Extending Nightwatch: custom assertions

```
var util = require('util');
exports.assertion = function(tagName, minCount, msg) {

  var defaultMessage = 'Testing if there are more than %s <s> elements on the page';
  var errorMessage = 'Error executing command';

  // Set default message
  this.message = msg || util.format(defaultMessage, minCount, tagName);

  // The expected text
  this.expected = function () {
    return 'to find at least ' + (minCount+1) + ' ' + tagName + ' elements on page';
  };

  // returning true means assertion passed
  // returning false means assertion failed
  this.pass = function(value) {
    return (value > minCount);
  };

  // returning true means element could not be found
  this.failure = function (result) {
    var failed = (result === false || (result && result.status === -1));

    if (failed) {
      this.message = msg || errorMessage;
    }

    return failed;
  };
};
```

learn more: nightwatchjs.org/guide#custom-assertions

Nightwatch.js

```
module.exports = {  
  src_folders: ['./tests'],  
  output_folder: './results',  
  custom_commands_path: './commands',  
  custom_assertions_path: './assertions',
```

```
  selenium: {  
    start_process: true,  
    server_path: './selenium-server-standalone-2.38.0.jar',  
    log_path: './results',  
    host: '127.0.0.1',  
    port: 4444  
  },
```

```
  test_settings: {  
    default: {  
      selenium_host: '127.0.0.1',  
      selenium_port: 4444
```



more to explore

before and after test hooks!

create custom commands!

take screenshots!

create custom assertions!

use sauce labs to run tests!

contribute to nightwatch!

use tags to organize test suites!

...and more

Get started

prerequisite: [node.js](#)

1. Use npm to install nightwatch
2. Download selenium-server-standalone.jar
3. Create Nightwatch config file
4. Create some tests



generator-selenium-nightwatch



Yeoman generator: bootstrap a Nightwatch.js selenium testing project.

```
$ npm install generator-selenium-nightwatch
```

Want to see pretty graphs? [Log in now!](#)

1 downloads in the last day
6 downloads in the last week
32 downloads in the last month

Last Published By



sethmcl

Version **0.0.6** last updated 4 months ago

License **MIT**

Keywords **yeoman-generator**

Repository **[git://github.com/sethmcl/generator-selenium-nightwatch](https://github.com/sethmcl/generator-selenium-nightwatch)** (git)

Homepage **<https://github.com/sethmcl/generator-selenium-nightwatch>**

Closing thoughts

Nightwatch.js is best selenium testing framework available to JavaScript developers

Good community, actively supported

Easy to get started, check it out

Resources

documentation

<http://nightwatchjs.org/guide> & <http://nightwatchjs.org/api>

sample code

<https://github.com/sethmcl/join-the-dark-side>

nightwatch generator (quick start)

<https://github.com/sethmcl/generator-selenium-nightwatch>

nightwatch page object model

<https://github.com/beatfactor/nightwatch/issues/242>