# What is Nightwatch

nightwatch v1.0.16 Weekly downloads 155k+ author Jagadeesh Shetty

Nightwatch.js is an automated testing framework for web applications and websites, written in Node.js and using the W3C WebDriver API (formerly Selenium WebDriver).

It is a complete End-to-End testing solution which aims to simplify writing automated tests and setting up Continuous Integration. Nightwatch can also be used for writing Node.js unit and integration tests.

*The name Nightwatch was inspired by the famous painting The Night Watch by Dutch artist Rembrandt van Rijn. The masterpiece is prominently displayed in the Rijksmuseum, in Amsterdam - The Netherlands.*

# Overview of WebDriver

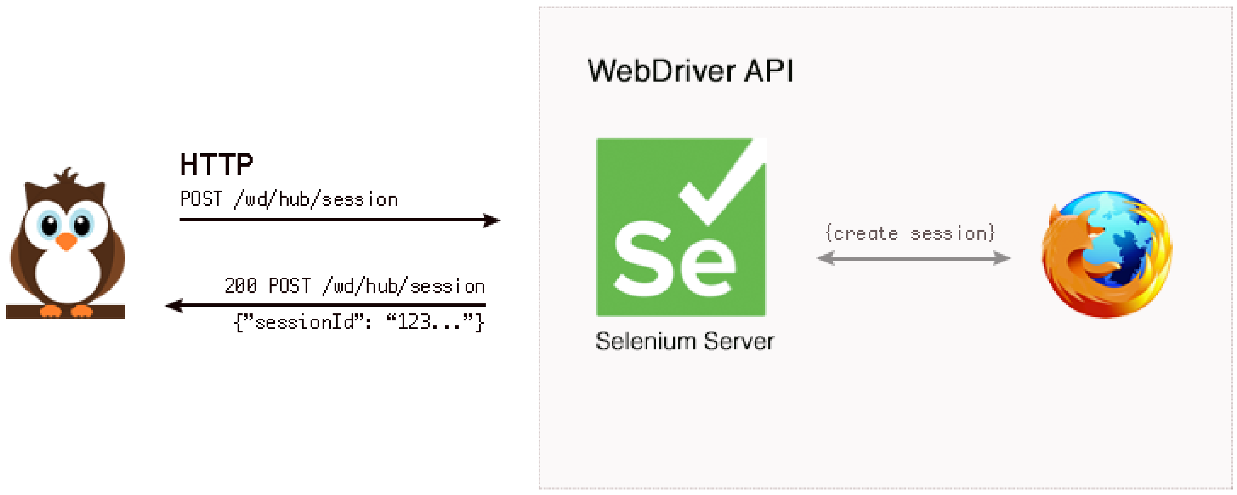
WebDriver is a general purpose library for automating web browsers. It was started as part of the Selenium project, which is a popular and comprehensive set of tools for browser automation initially written for Java but now with support for most programming languages.

Nightwatch uses the WebDriver API to perform the browser automation related tasks, like opening windows and clicking links for instance.

WebDriver is now a W3C specification aiming to standardize browser automation. WebDriver is a remote control interface that enables introspection and control of user agents. It provides a platform and a restful HTTP api as a way for web browsers to be remotely controlled.

# Theory of Operation

Nightwatch works by communicating over a restful HTTP API with a WebDriver server (such as ChromeDriver or Selenium Server). The protocol is defined by the W3C WebDriver spec, which is derived from JSON Wire protocol. See below for an example workflow for browser initialization.



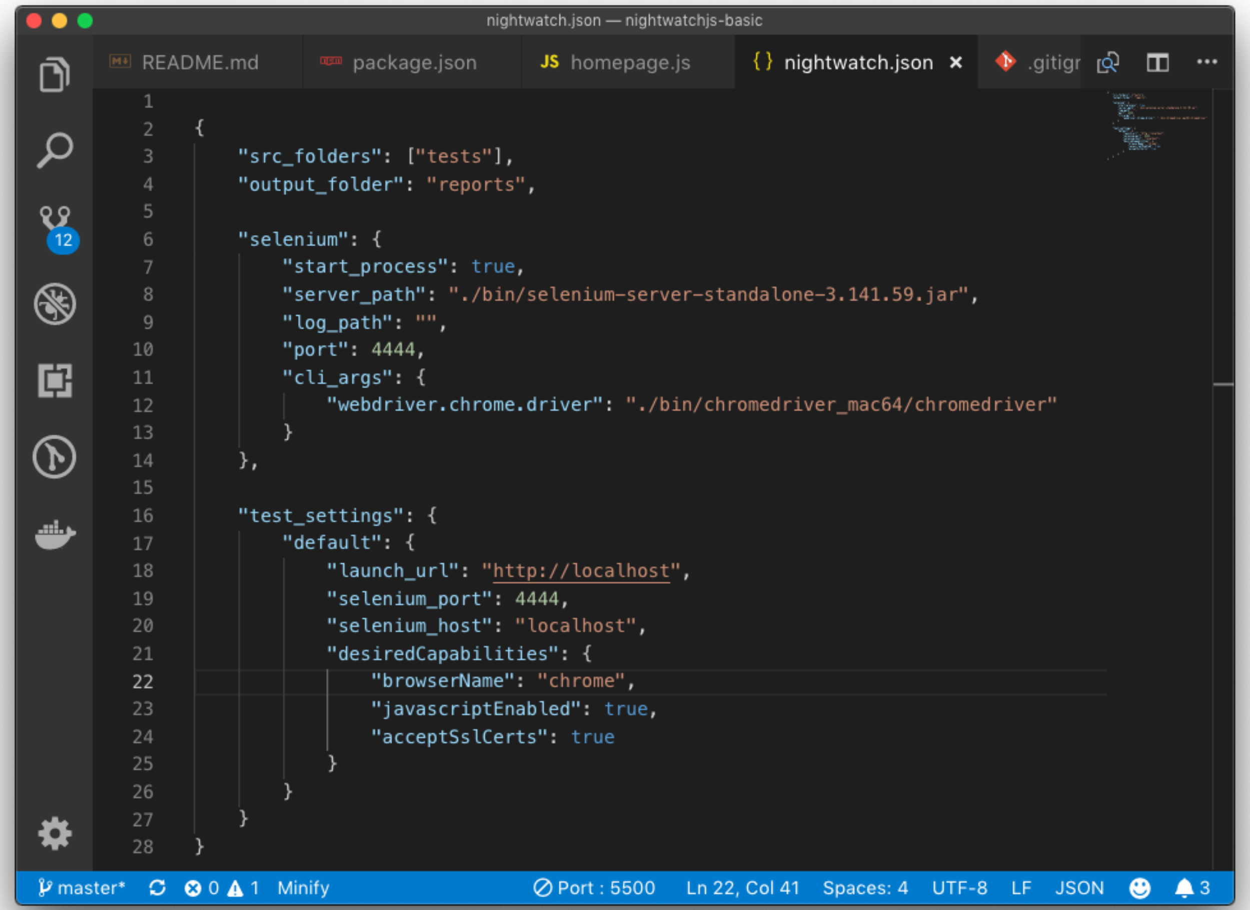
Most of the times, Nightwatch needs to send at least 2 requests to the WebDriver server in order to perform a command or assertion. ...The first one being the request to locate an element given a CSS selector (or Xpath expression) ...Next to perform the actual command/assertion on the given element.

## Setup

1. **Node.js** should be installed.
2. **NPM** should be installed.
3. Create a directory with **nightwatchjs-basic** name.
4. Run `npm init` and complete with basic details.
5. Run `npm install nightwatch` to install nightwatch within workspace.
6. Download `selenium` and `chrome` drivers and copy to `bin` directory.
7. Create `nightwatch.json` file within working directory. The nightwatch test runner binary expects a configuration file.
8. Create `tests` directory and create `homepage.js` file.
9. Update `package.json` with test runner for `scripts` value.

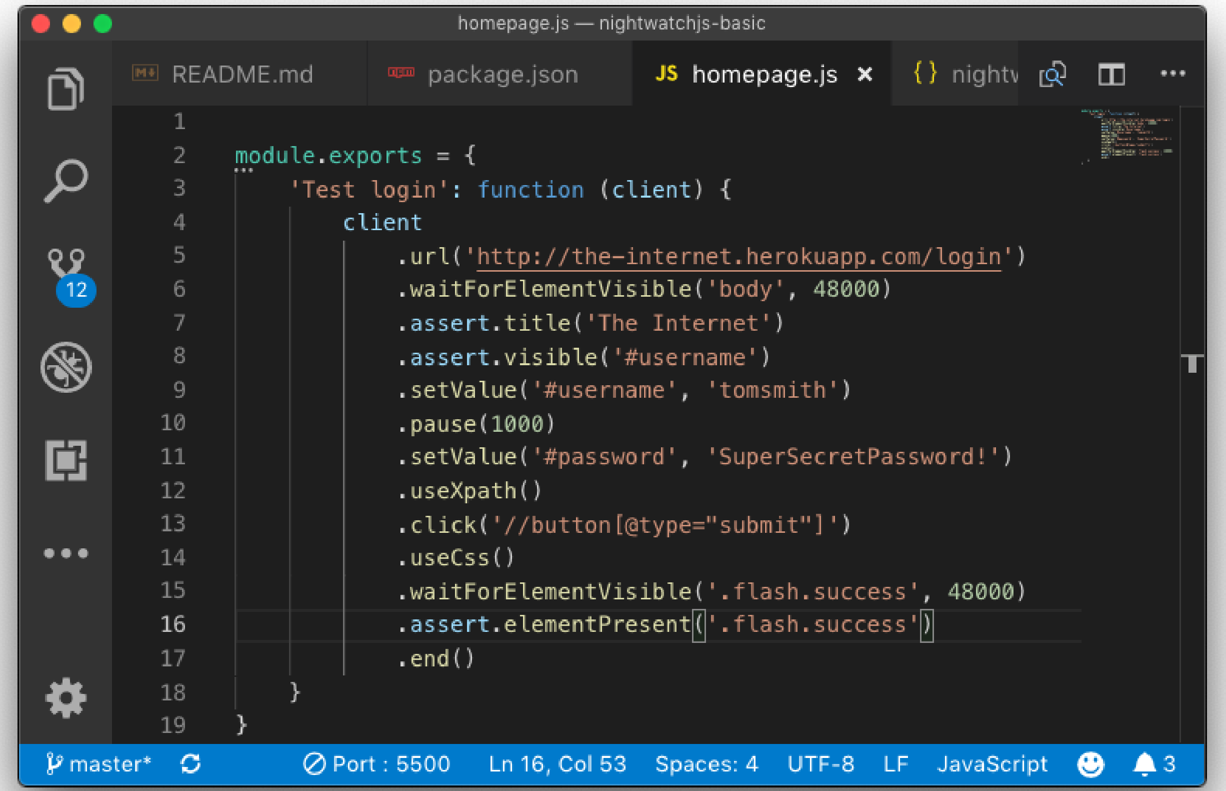
## Screenshots

### nightwatch.conf



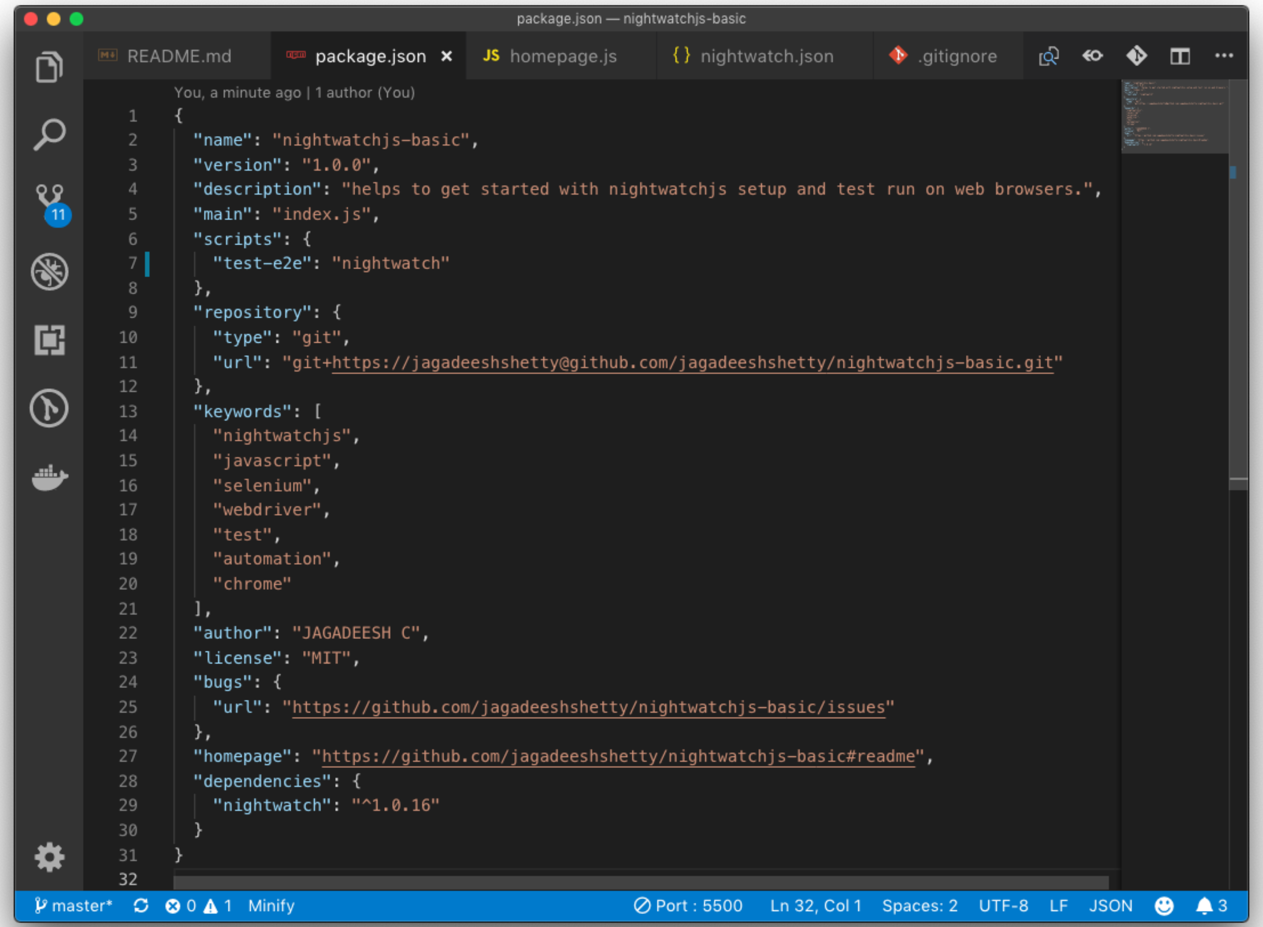
```json

{
    "src_folders": ["tests"],
    "output_folder": "reports",

    "selenium": {
        "start_process": true,
        "server_path": "./bin/selenium-server-standalone-3.141.59.jar",
        "log_path": "",
        "port": 4444,
        "cli_args": {
            "webdriver.chrome.driver": "./bin/chromedriver_mac64/chromedriver"
        }
    },

    "test_settings": {
        "default": {
            "launch_url": "http://localhost",
            "selenium_port": 4444,
            "selenium_host": "localhost",
            "desiredCapabilities": {
                "browserName": "chrome",
                "javascriptEnabled": true,
                "acceptSslCerts": true
            }
        }
    }
}
```

## tests/homepage.js



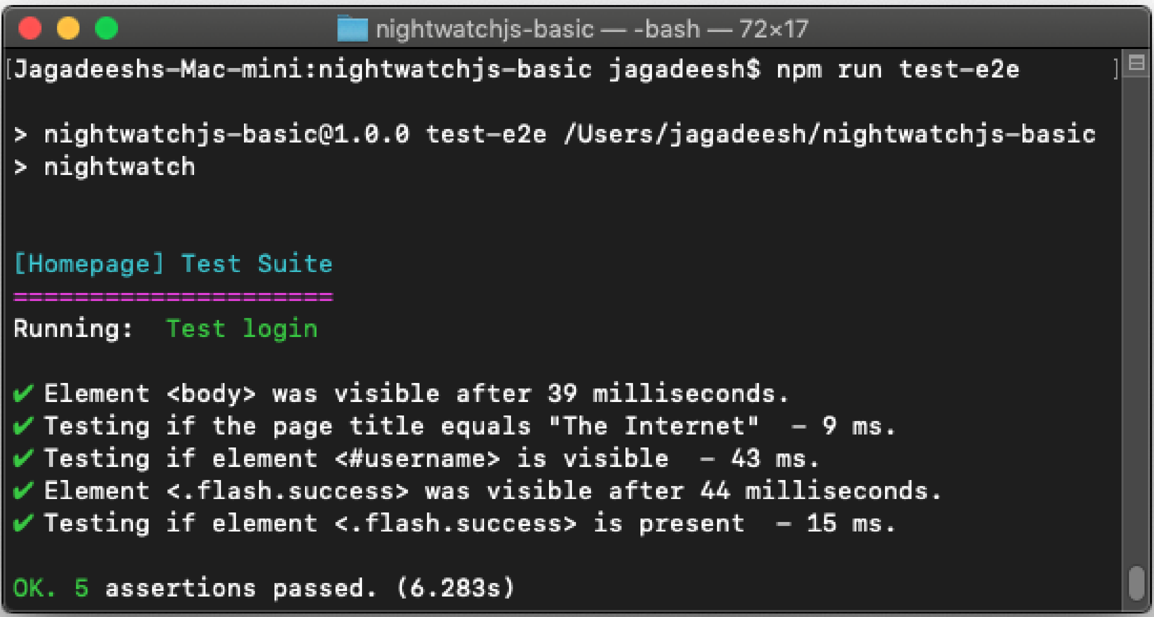
```

module.exports = {
    'Test login': function (client) {
        client
            .url('http://the-internet.herokuapp.com/login')
            .waitForElementVisible('body', 48000)
            .assert.title('The Internet')
            .assert.visible('#username')
            .setValue('#username', 'tomsmith')
            .pause(1000)
            .setValue('#password', 'SuperSecretPassword!')
            .useXpath()
            .click('//button[@type="submit"]')
            .useCss()
            .waitForElementVisible('.flash.success', 48000)
            .assert.elementPresent('.flash.success')
            .end()
    }
}
```

## package.json



```json
{
  "name": "nightwatchjs-basic",
  "version": "1.0.0",
  "description": "helps to get started with nightwatchjs setup and test run on web browsers.",
  "main": "index.js",
  "scripts": {
    "test-e2e": "nightwatch"
  },
  "repository": {
    "type": "git",
    "url": "git+https://jagadeeshshetty@github.com/jagadeeshshetty/nightwatchjs-basic.git"
  },
  "keywords": [
    "nightwatchjs",
    "javascript",
    "selenium",
    "webdriver",
    "test",
    "automation",
    "chrome"
  ],
  "author": "JAGADEESH C",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/jagadeeshshetty/nightwatchjs-basic/issues"
  },
  "homepage": "https://github.com/jagadeeshshetty/nightwatchjs-basic#readme",
  "dependencies": {
    "nightwatch": "^1.0.16"
  }
}
```

## Run



```
Jagadeeshs-Mac-mini:nightwatchjs-basic jagadeesh$ npm run test-e2e

> nightwatchjs-basic@1.0.0 test-e2e /Users/jagadeesh/nightwatchjs-basic
> nightwatch


[Homepage] Test Suite
=====================
Running:  Test login

✔ Element <body> was visible after 39 milliseconds.
✔ Testing if the page title equals "The Internet"  - 9 ms.
✔ Testing if element <#username> is visible  - 43 ms.
✔ Element <.flash.success> was visible after 44 milliseconds.
✔ Testing if element <.flash.success> is present  - 15 ms.

OK. 5 assertions passed. (6.283s)
```

## Reference

Official Nightwatch