**Updated:** 5/2/2020 4:40 PM

# Overview

* Provides a command line interface for aXe to run quick accessibility tests.
* By default, axe-cli runs Chrome in headless mode, which requires Chrome 59 or up
* Install axe-cli globally
* Install the web drivers of the browser(s) you with to use. A web driver is a driver for your web browsers. It allows other programs on your machine to open a browser and operate it.

# Software required

* Node.js 6+
  + <https://nodejs.org/en/>
* Browser Webdriver(s)
  + Chrome: <https://chromedriver.chromium.org/home>
  + Firefox: <https://github.com/mozilla/geckodriver/releases>
* Axe-core
  + <https://github.com/dequelabs/axe-cli>
  + Update to most recent axe-core library:
    - npm install axe-core --save-dev
  + <https://www.deque.com/blog/axe-updates-deque-releases-axe-core-3-5/>

# Installation

* Node.js 6 or higher
* Chrome or Firefox Browser
* Browser Webdriver to match desired headless browser for testing
  + Download and install webdriver executable (chromedriver.exe, geckodriver.exe, etc.) If using something other than Chrome, that browser's webdriver must be available on your PATH.
* Axe-core globally:
  + npm install axe-cli -g
  + npm install axe-core --save-dev

# Usage and Syntax

## Test Single Page

After installing, you can now run the axe command in your CLI, followed by the URL of the page you wish to test:

* axe <webpage>

## Test multiple pages

You can run multiple pages at once, simply add more URLs to the command. Keep in mind that axe-cli is not a crawler, so if you find yourself testing dozens of pages at once, you may want to consider switching over to something like axe-webdriverjs. If you do not specify the protocol, http will be used by default:

* axe <page1>, <page2> (Note that there is a space after the comma in the syntax)

Run against code on local machine on the file system (not hosted by server)

* axe <filesystem path>/<filename>   
  **axe /c/\_test/TC1005C001.html**

## Specify rules

* axe <webpage> --rules <rule1>,<rule2>,<rule3> (Note that there is no space after the comma in the syntax)

## Specify tags

Run using all rules with a specified tag (see tag listing below)

* axe <webpage> --tags wcag2a (note the tags are case-sensitive and cannot be combined with the --rules tag)

## Specify browser

* axe <webpage> --browser <browser> (Note that any browser other than chrome must have a webdriver installed, and the folder location of that driver executable must be included in the PC environment’s PATH statement)

## Specify send results to JSON file

Results can be saved as JSON data, using the --save and --dir flags. By passing a filename to --save you indicate how the file should be called. If no filename is passed, a default will be used:

* axe <webpage> --save <JSON file name>
* axe <webpage> --dir <folder location, i.e. “./results/”>

## Specify send results to STDOUT

To output the test results to STDOUT, provide the --stdout flag. This flag has the side-effect of silencing all other logs/output (other than errors, which are written to STDERR).

* To print the entire result object to your terminal:
  + axe --stdout <url>
* To pipe the results to a file:
  + axe --stdout www.deque.com > your\_file.json
* To pipe the results to a JSON-parsing program for further processing, do:
  + axe --stdout www.deque.com | jq ".[0].violations"

## Defining the scope of a test

If you want to only test a specific area of a page, or wish to exclude some part of a page you can do so using the --include and --exclude flags and pass it a CSS selector:

* axe www.deque.com --include "#main" --exclude "#aside"
* You may pass multiple selectors with a comma-delimited string. For example:
  + axe www.deque.com --include "#div1,#div2,#div3"

## Custom axe-core versions

Axe-cli will look for locally available versions of axe-core. If the directory from where you start axe-cli has an axe.js file, or has a node\_modules directory with axe-core installed in it. Axe-cli will use this version of axe-core instead of the default version installed globally.

To specify the exact file axe-core file axe-cli should use, you can use the --axe-source flag (-a for short), with a relative or absolute path to the file.

* axe www.deque.com --axe-source ./axe.nl.js

## Custom Chrome Flags

When using the Headless Chrome browser, you may provide any number of flags to configure how the browser functions.

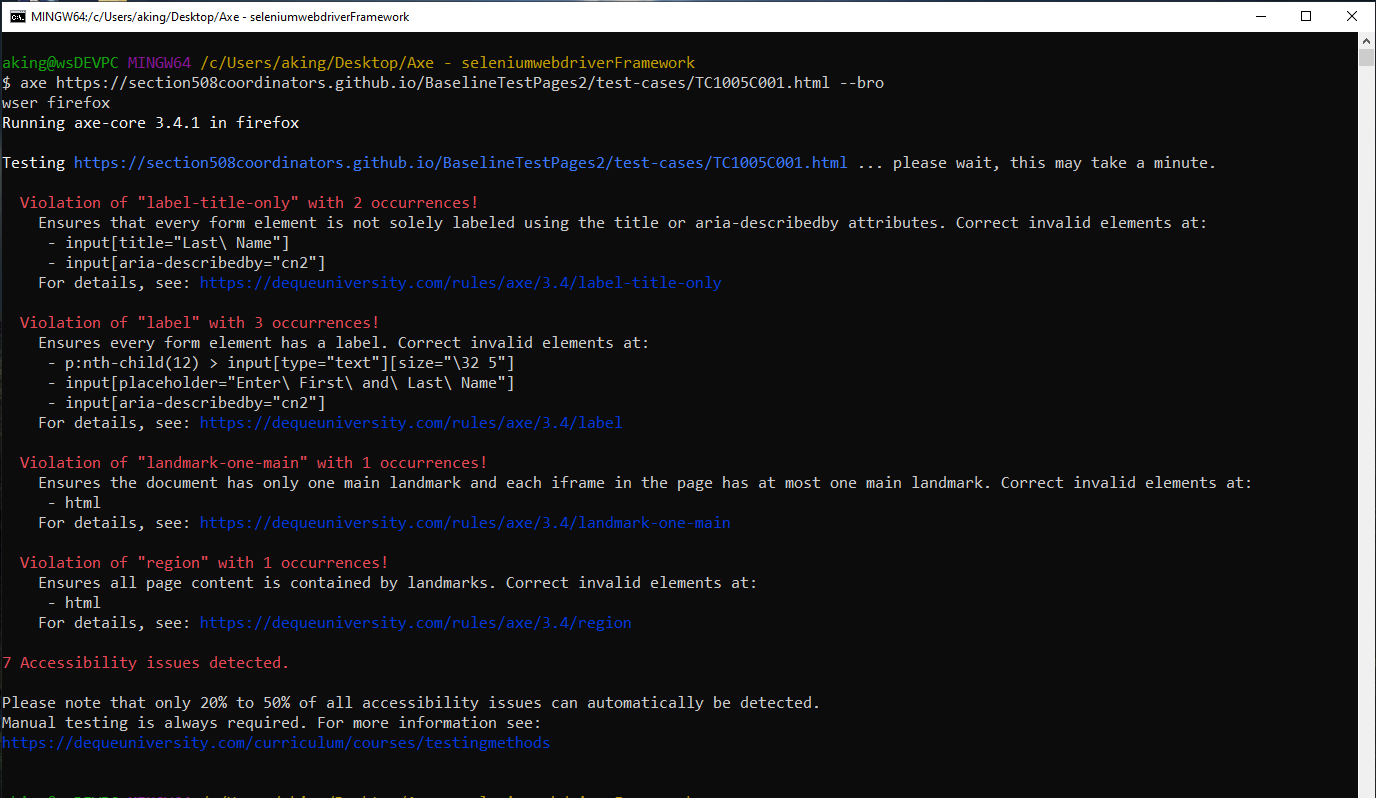
Options are passed by name, without their leading -- prefix. For example, to provide the --no-sandbox --disable-setuid-sandbox --disable-dev-shm-usage flags to the Chrome binary, you'd do:

* axe --chrome-options="no-sandbox,disable-setuid-sandbox,disable-dev-shm-usage" [www.deque.com](http://www.deque.com)

# Examples

## Axe-CLI-Command-Ex-01

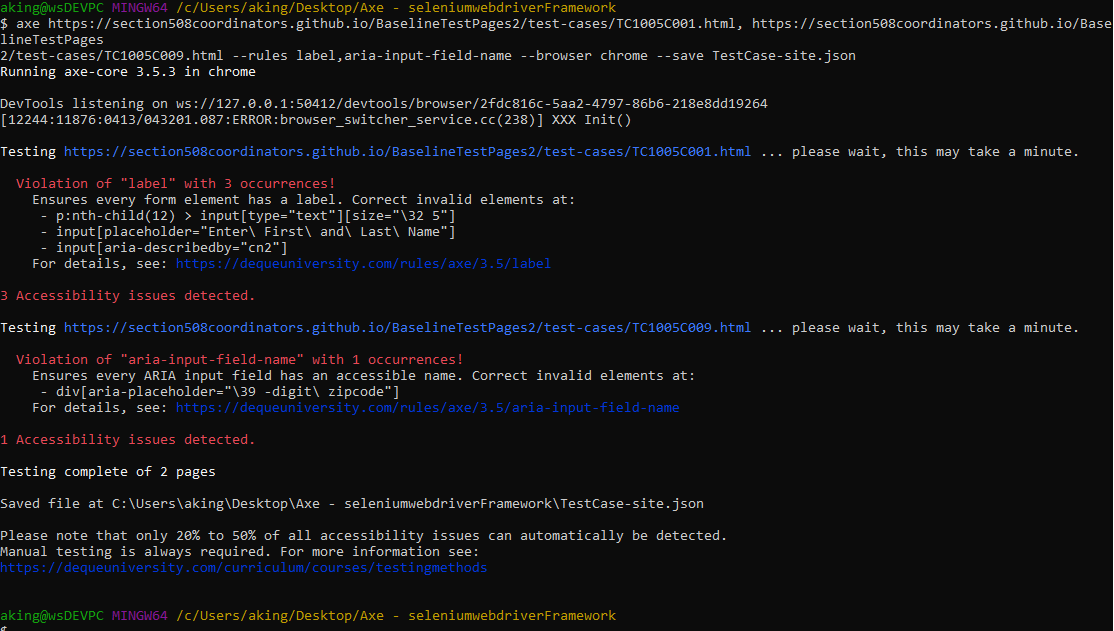
One page, Firefox browser, default ruleset

axe https://section508coordinators.github.io/BaselineTestPages2/test-cases/TC1005C001.html --browser firefox  
***Console window output:***

## Axe-CLI-Command-Ex-02

Multi-page, chrome browser, specific rules only, save results to JSON file

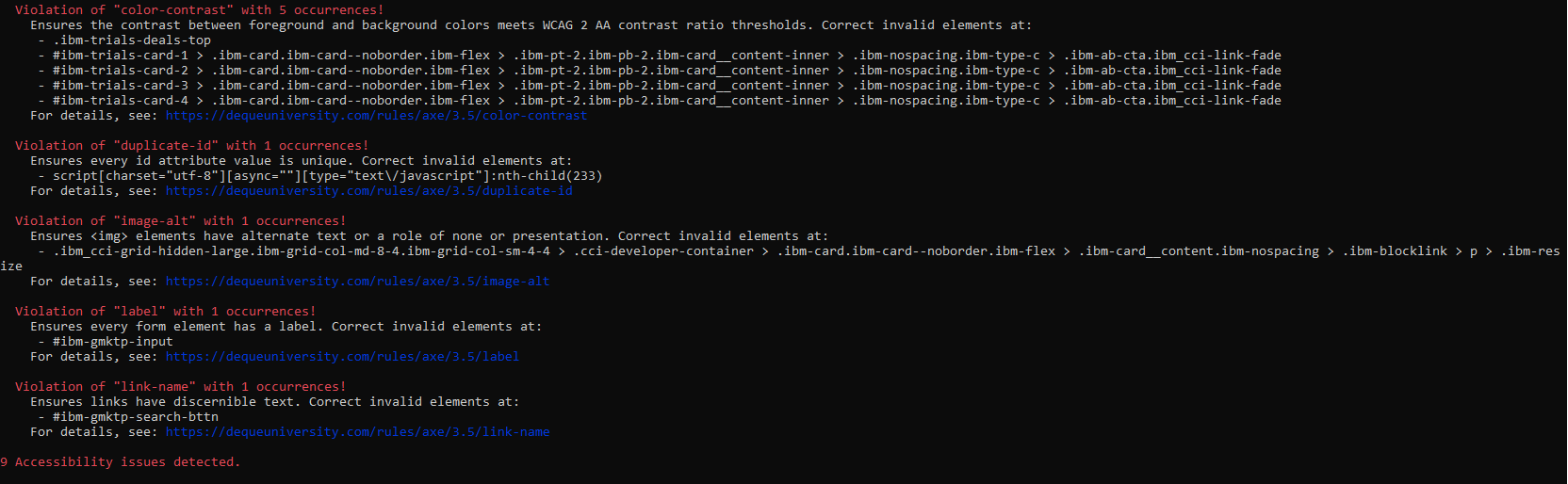
axe https://section508coordinators.github.io/BaselineTestPages2/test-cases/TC1005C001.html, https://section508coordinators.github.io/BaselineTestPages2/test-cases/TC1005C009.html --rules label,aria-input-field-name --browser chrome --save TestCase-site.json

***Console window output:***

## axe-CLI-Command-Ex-03

One-page test with OAST recommended rules only

axe www.ibm.com --rules scrollable-region-focusable,role-img-alt,image-alt,color-contrast,aria-hidden-focus,aria-input-field-name,aria-toggle-field-name,label,form-field-multiple-labels,document-title,td-headers-attr,empty-heading,listitem,link-name,input-image-alt,input-button-name,button-name,html-has-lang,html-lang-valid,valid-lang,frame-title,duplicate-id

***Console window output:***

# Tips

1. Always be aware of the currently published version of the axe-core library. If you want to run with the most current version – which will likely have you up to date on the latest rules and bug fixes, be sure to periodically update your copy of the library and note the axe-core version message when running CLI: “*Running axe-core <library version> in <selected browser*>”
2. Be wary of using the --tags option. Results obtained by OAST through testing are questionable. OAST has recommended individual rules in favor of using tags.

# Appendix

## Tags Listing

| **Tag Name** | **Accessibility Standard/Purpose** |
| --- | --- |
| wcag2a | WCAG 2.0 & WCAG 2.1 Level A |
| wcag2aa | WCAG 2.0 & WCAG 2.1 Level AA |
| wcag21a | WCAG 2.1 Level A |
| wcag21aa | WCAG 2.1 Level AA |
| section508 | Section 508 |
| best-practice | Best practices endorsed by Deque |
| experimental | Cutting-edge techniques |
| cat | Category mappings used by Deque (see below) |