

# Capítulo 5 - Explorando o Axe-CLI

---

Neste capítulo, vou mostrar como usar o Axe-CLI para que você possa detectar rapidamente os problemas básicos de acessibilidade de seu aplicativo.

## Por que usar Axe-CLI?

---

Primeiro, você pode integrá-lo facilmente em seu pipeline de integração contínua.

É rápido, o que você verá na demonstração mais tarde, e também é fácil de instalar e configurar.

Por último, nenhum código é necessário, portanto, oferece a opção de detectar problemas de acessibilidade com antecedência, sem escrever código adicional.

## Pré requisitos:

---

Como pré-requisitos, para instalar o Axe-CLI, você precisa ter:

- Uma IDE instalada e configurada
- Node e o NPM instalados em sua máquina.

Se você não os instalou, vá em frente e volte quando tudo estiver pronto.

Se quiser saber se o Node foi instalado com sucesso, basta digitar:

```
node -v
```

Você pode consultar os recursos abaixo para encontrar o link sobre como instalar o Node e o NPM.

## Instalando o Axe-CLI

---

Para instalar o Axe-CLI, tudo o que precisamos é este comando:

```
npm install -g axe-cli
```

O **-g** significa global, o que indica que vamos adicionar este módulo globalmente.

## Executando Demonstração Axe-CLI

---

Assim que terminar, você pode digitar o comando **axe** em seu terminal e ele deve fornecer a seguinte saída:

```
Running axe-core 4.1.2 in chrome-headless  
No url was specified. Check `axe -h` for help
```

Ele está nos dando um erro porque não fornecemos nenhum URL para verificação.

Vamos prosseguir e digitar **axe** novamente, seguido de qualquer URL que você deseje.

Para esta demonstração, forneci o URL do meu blog pessoal:

```
axe mariedrake.com
```

O terminal fornecerá a seguinte saída:

```
Running axe-core 4.1.2 in chrome-headless
```

```
Testing http://mariedrake.com ... please wait, this may take a minute.  
0 violations found!
```

```
Please note that only 20% to 50% of all accessibility issues can automatically be  
detected.
```

```
Manual testing is always required. For more information see:  
https://dequeuniversity.com/curriculum/courses/testingmethods
```

Por padrão, isso executará seus testes no Chrome headless.

Quando terminar, ele fornecerá a saída padrão aqui mostrando quaisquer erros, se houver, bem como um link para como corrigir o problema.

Parece que Axe não relatou nenhuma violação em meu blog pessoal, o que é muito bom.

No entanto, é importante notar que só porque Axe não detectou nenhum problema, não significa que meu site está totalmente acessível.

Eu gosto que Axe adicionou esta mensagem aqui que "apenas de 20% a 50% dos problemas de acessibilidade são detectados".

Portanto, o teste de acessibilidade real ainda é necessário.

Se você deseja executar o teste em um navegador real, o que você pode fazer é passar o parâmetro do navegador seguido do nome do navegador.

```
axe mariedrake.com --browser chrome
```

Portanto, neste caso, é o Chrome.

Execute o teste novamente e, desta vez, você verá que o navegador Chrome foi aberto e será fechado quando a execução do teste for concluída.

Se você deseja executar o teste em várias páginas, pode adicionar mais URLs a ele, contanto que sejam separados por vírgula.

Vamos executar este teste também na minha página **sobre** e ver se há algum problema.

```
axe mariedrake.com, mariedrake.com/about --browser chrome
```

```
Running axe-core 4.1.2 in chrome-headless
```

Testing <http://mariedrake.com/about> ... please wait, this may take a minute.

Violation of "heading-order" with 1 occurrences!

Ensures the order of headings is semantically correct. Correct invalid elements at:

- #comp-kchrofdz > h6

For details, see: <https://dequeuniversity.com/rules/axe/4.1/heading-order>

Violation of "link-name" with 2 occurrences!

Ensures links have discernible text. Correct invalid elements at:

- #dataItem-k2wkcncb2-comp-k2wc00fa > .fNVUF[href\$="mcruzdrake"][target="\_blank"]
- #dataItem-k2wkcncb2-comp-k2wc00fa > .fNVUF[target="\_blank"][rel="noopener"]

For details, see: <https://dequeuniversity.com/rules/axe/4.1/link-name>

Violation of "object-alt" with 1 occurrences!

Ensures <object> elements have alternate text. Correct invalid elements at:

- object

For details, see: <https://dequeuniversity.com/rules/axe/4.1/object-alt>

4 Accessibility issues detected.

Estou vendo mais erros aqui agora, o que significa que meu blog não está em conformidade com algumas das diretrizes.

Neste exemplo, desejo desabilitar a regra do **link-name**.

```
axe mariedrake.com/about --disable link-name
```

Quando a execução for concluída, você verá apenas os outros erros relatados e excluirá aqueles que acabamos de desativar.

Running axe-core 4.1.2 in chrome-headless

Testing <http://mariedrake.com/about> ... please wait, this may take a minute.

Violation of "heading-order" with 1 occurrences!

Ensures the order of headings is semantically correct. Correct invalid elements at:

- #comp-kchrofdz > h6

For details, see: <https://dequeuniversity.com/rules/axe/4.1/heading-order>

Violation of "object-alt" with 1 occurrences!

Ensures <object> elements have alternate text. Correct invalid elements at:

- object

For details, see: <https://dequeuniversity.com/rules/axe/4.1/object-alt>

2 Accessibility issues detected.

Se você não tiver certeza de qual é o nome da regra, apenas copie e cole o nome da regra que você pode ver aqui como parte da violação.

Então, por exemplo, se eu quiser desabilitar a regra **heading-order**, eu só preciso fornecer a ordem do título no parâmetro **--disable**.

Agora, imagine se você tiver muitos erros.

Você provavelmente deseja enviar a saída para um arquivo JSON em algum lugar, em vez de deixá-lo em seu terminal.

A boa notícia é que o Axe-CLI permite que você faça isso facilmente.

Basta passar o parâmetro **--save** seguido pelo nome do arquivo para o seu arquivo JSON.

Portanto, para este exemplo, vamos chamar o nome do arquivo **test-results.json**.

```
axe marriedrake.com --save test-results.json
```

Agora, quando a execução do teste for concluída, você verá um arquivo JSON criado no final, que mostra a saída geral da execução do teste.

```
Running axe-core 4.1.2 in chrome-headless

Testing http://marriedrake.com ... please wait, this may take a minute.

Violation of "link-name" with 2 occurrences!
Ensures links have discernible text. Correct invalid elements at:
  - #dataItem-k2wkcncb2-comp-k2wc00fa > .fNVUF[href$="mcruzdrake"][target="__blank"]
  - #dataItem-k2wkcnc-k-comp-k2wc00fa > .fNVUF[target="__blank"][rel="noopener"]
For details, see: https://dequeuniversity.com/rules/axe/4.1/link-name

2 Accessibility issues detected.

Saved file at /Users/user/RepositorioUser/user-ally/test-results.json
```

Você pode encontrar as informações aqui sobre quais regras são inaplicáveis, ou você também pode pesquisar as violações a partir daqui, e os erros serão relatados neste formato JSON.

Para terminar este capítulo, tenho uma pequena atividade para você se sentir confortável com o uso do Axe-CLI.

Escolha qualquer site de sua escolha e use o Axe-CLI para encontrar quaisquer problemas de acessibilidade.

Você pode usar os diferentes parâmetros que viu na demonstração como um guia.

Também adicionei um link para a página Axe-CLI GitHub, que você também pode usar como referência.

Quando terminar, iremos para o próximo capítulo e veremos como escrever testes de acessibilidade automatizados com Cypress e Cypress-axe.

## Links

<https://github.com/dequelabs/axe-cli>

<https://nodejs.org/en/download/>