

Hello 🖐️

Yuraima Estevez



Tech Lead



The New York Times

Accessibility Testing Automation

Accessibility Testing Automation

How to do Good, Without doing Anything

Tools Testing

Tools Testing

Tools

eslint-plugin-jsx-ally

Set of accessibility linter rules that checks JSX elements during development

Lighthouse

An open-source, automated tool for improving the quality of web pages.

Lighthouse

- Chrome DevTools
- Command Line
- CI Systems
- Web UI


useReactNYC

usereact.nyc

NYT Internal Gmail Calendar My Tickets vi GCP Scout Stage Scout Local local stage Scoop Local Login HIKE: Our quick, s... Bytes 52 Places to Go in... Other E

Elements Console Sources Network Performance Audits >> 2

new audit)



Generate report

Identify and fix common problems that affect your site's performance, accessibility, and user experience.
[Learn more](#)

Categories


☐ Performance
☐ Progressive Web App
☐ Best practices
☒ Accessibility
☐ SEO

Device

☐ Mobile
☒ Desktop

Community Plugins(beta)

☐ Publisher Ads



Welcome to useReactNYC

The Home of New York City's Newest and Most Exciting React Community Meetup

useReactNYC

useReactNYC

NYT InternalGmailCalendarMy TicketsviGCPScout StageScout LocallocalstageScoop Local LoginHIKE: Our quick, s...Bytes52 Places to Go in...Other E

ElementsConsoleSourcesNetworkPerformanceAudits2

12:05 PM - useReactNYC

https://useReactNYC/

96

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

— These are opportunities to improve the legibility of your content.

Background and foreground colors do not have a sufficient contrast ratio.

— These are opportunities to improve the semantics of the controls in your application. This enhances the experience for users of assistive technology, like a screen reader.

name> or <iframe> elements do not have a title


Additional items to manually check (11) — These items address areas which an automated testing tool cannot detect. Learn more in our guide on [conducting an accessibility review](#).

Audits (20)

Fixable (13)

Runtime Settings

https://useReactNYC/



Welcome to useReactNYC

The Home of New York City's Newest and Most Exciting React Community Meetup

useReactNYC

useReactNYC

NYT InternalGmailCalendarMy TicketsviGCPScout StageScout LocallocalstageScoop Local LoginHIKE: Our quick, s...Bytes52 Places to Go in...Other E

ElementsConsoleSourcesNetworkPerformanceAudits2

12:05 PM - useReactNYC

https://useReactNYC/

96

Accessibility


These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

— These are opportunities to improve the legibility of your content.

Background and foreground colors do not have a sufficient contrast ratio.

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

Rendering Elements



Welcome to useReactNYC

The Home of New York City's Newest and Most Exciting React Community Meetup

useReactNYC

useReactNYC

NYT InternalGmailCalendarMy TicketsviGCPScout StageScout LocallocalstageScoop Local LoginHIKE: Our quick, s...Bytes52 Places to Go in...Other E

ElementsConsoleSourcesNetworkPerformanceAudits2

07:58 AM - useReact.nyc

ps://useReact.nyc/

67

Performance

96

Accessibility

93

Best Practices

100

SEO

0-49

50-89

90-100

were issues affecting this run of Lighthouse:

chrome extensions negatively affected this page's load performance. Try auditing the page in incognito mode or m a Chrome profile without extensions.

67

Performance

Contentful Paint3.3 s

First Meaningful Paint3.3 s


ed Index5.6 s

First CPU Idle5.0 s

e to Interactive5.3 s

Max Potential First Input Delay870 ms

Trace



Welcome to useReactNYC

The Home of New York City's Newest and Most Exciting React Community Meetup

react-axe

Open source tool that tests the accessibility of React applications and displays the results in the Chrome DevTools console.

ChromeFileEditViewHistoryBookmarksPeopleWindowHelp

NYT Programs – 30-Day Well C X +

localhost.nytimes.com:3001/programs/well-challenge

AppsNYT InternalGmailCalendarMy Ticketsguides-wellprogramsguides-adminvi guides-apiBytesScout StageAdmin localOther Bookmarks

ElementsNetworkSecurityAuditsConsole>>72

topaxeDefault levels227 hidden

New aXe issuesindex.js:146

minor: id attribute value must be uniqueindex.js:166

https://dequeuniversity.com/rules/axe/3.2/duplicate-id?application=axeAPI

moderate: Heading levels should only increase by oneindex.js:166

https://dequeuniversity.com/rules/axe/3.2/heading-order?application=axeAPI

serious: <html> element must have a lang attributeindex.js:166

https://dequeuniversity.com/rules/axe/3.2/html-has-lang?application=axeAPI

moderate: Document must have one main landmarkindex.js:166

https://dequeuniversity.com/rules/axe/3.2/landmark-one-main?application=axeAPI

moderate: All page content must be contained by landmarksindex.js:166

https://dequeuniversity.com/rules/axe/3.2/region?application=axeAPI

New aXe issuesindex.js:146

serious: Elements must have sufficient color contrastindex.js:166


https://dequeuniversity.com/rules/axe/3.2/color-contrast?application=axeAPI

The New York Times

30-Day Well Challenge

Take a small step every day for 30 days to build healthy habits for your body, mind and spirit.


GETTING READY



PREPARE




Write Your Way to Success

ADVERTISEMENT



Empower Your Business

New **Lenovo 14w** is an ideal notebook at an ideal price

[LEARN MORE](#)

a11y-testing.yuraima.com

@yurm04

axe-core

Accessibility engine for automated Web UI testing

Testing

Testing

Unit Tests
Integration Tests
End-to-end Tests

Unit Tests

Component API and Behavior
ARIA attributes
Focus States

Jest react-testing-library

```
const AccessibleButton = ({
  disabled = false,
  children,
  type = "button",
  handleClick,
  accessibleLabel
}) => (
  <button
    disabled={disabled}
    type={type}
    onClick={handleOnClick}
    data-testid='button-el'
    // only add an aria-label if an accessibleLabel is passed in props
    {...accessibleLabel && { 'aria-label': accessibleLabel }}
  >
    {children}
  </button>
)
```

```
it('does NOT render "aria-label" if accessibleLabel is not passed', () => {
  const button = (
    <AccessibleButton>
      This button is accessible
    </AccessibleButton>
  )
  const { container } = render(button)
  const renderedButton = getByTestId(container, 'button-el')

  expect(renderedButton).not.toHaveAttribute('aria-label')
})
```



```
it('has the "aria-label" attribute set if an accessibleLabel is passed', () => {
  const ariaLabel = "Submit email to newsletter"
  const button = (
    <AccessibleButton
      accessibleLabel={ariaLabel}>
        👍
    </AccessibleButton>
  )
  const { container } = render(button)
  const renderedButton = getByTestId(container, 'button-el')

  expect(renderedButton).toHaveAttribute('aria-label', ariaLabel)
})
```

```
it('can have focus', () => {
  const button = (
    <AccessibleButton>
      This button is accessible
    </AccessibleButton>
  )
  const { container } = render(button)
  const renderedButton = getByTestId(container, 'button-el')

  renderedButton.focus()
  expect(renderedButton).toHaveFocus()
})
```

```
const AccessibleButton = ({
  disabled = false,
  children,
  type = "button",
  handleClick,
  accessibleLabel
}) => (
  <div
    disabled={disabled}
    type={type}
    onClick={handleOnClick}
    data-testid='button-el'
    // only add an aria-label if an accessibleLabel is passed in props
    {...accessibleLabel && { 'aria-label': accessibleLabel }}
  >
    {children}
  </div>
)
```

FAIL src/components/AccessibleButton/__tests__/index.js

AccessibleButton

✗ can have focus (29ms)

• AccessibleButton > can have focus

expect(element).toHaveFocus()

Expected

`<div data-testid="button-el" type="button">This button is accessible</div>`

Received:

`<body><div><div data-testid="button-el" type="button">This button is accessible</div></div></body>`

```
56 |  
57 |     renderedButton.focus()  
> 58 |     expect(renderedButton).toHaveFocus()  
    |                               ^  
59 |   })  
60 | })  
61 |
```

at Object.it (src/components/AccessibleButton/__tests__/index.js:58:28)

Test Suites: 1 failed, 1 total

Tests: 1 failed, 1 total

Snapshots: 0 total

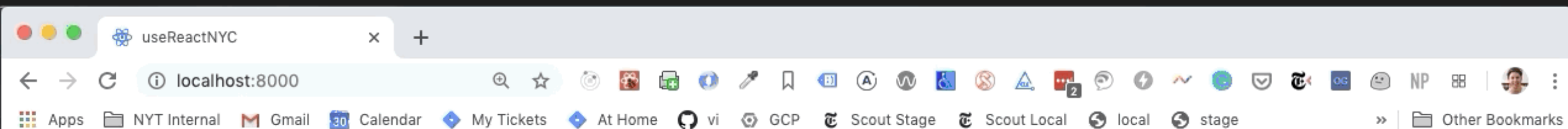
Time: 2.493s

Ran all test suites related to changed files.

Watch Usage: Press w to show more._

Integration Tests

How individual components interact with each other

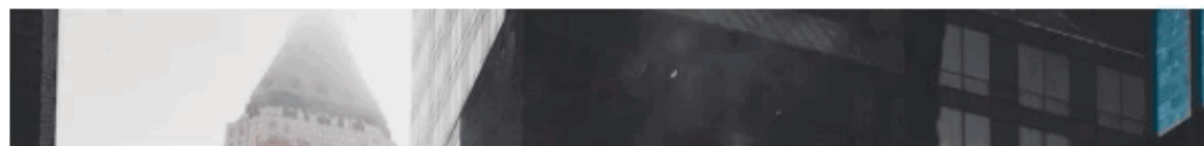


staff will be happy to help participants contact venue security or local law enforcement, provide escorts, or otherwise assist those experiencing harassment to feel safe for the duration of events. We value your attendance.

We expect participants to follow these rules at all conference venues and conference-related social events.

Enter your email to join our Slack!

Resources



```
    <div>Enter your email to receive our blog! </div>
    <AccessibleInput
      type="email"
      name="email"
      placeholder="yuraima@useReact.com"
      onBlur={validateEmailAddress}
      className={showError ? 'error' : ''}
    />
    <AccessibleButton
      accessibleLabel="Submit email address"
      disabled={!isValidEmail}

      handleClick={submitEmail}>
      Submit
    </AccessibleButton>
    <AccessibleAlert
      message={showError && errorMessage} />
  </form>
)
}
```

```
const AccessibleAlert = ({ message = '' }) => (  
  <div className="visually-hidden">  
    <p role="alert" data-testid="alert-message">{message}</p>  
  </div>  
)
```



```
describe('SlackSignupContainer', () => {

  // ... other tests

  it('renders visibly hidden alert on input blur without valid email', () => {
    const errorMessage = 'Please enter a valid email address'
    const { container } = render(<SlackSignup errorMessage={errorMessage} submitEmail={() => {}} />)
    const renderedInput = getByTestId(container, "input")
    const renderedAlert = getByTestId(container, "alert-message")

    /**
     * there should be no error message when the slack
     * sign up component renders
     */
    renderedInput.focus()
    expect(renderedAlert).not.toHaveTextContent()

    /**
     * when we blur without a valid email address the
     * alert component should now contain our error message
     */
    renderedInput.blur()
    expect(renderedAlert).toHaveTextContent(errorMessage)
  })

  // ... other tests

})
```

End-to-end Tests

Evaluate your entire app and full user flows

cypress.io

cypress-axe plugin

```
describe("Accessibility tests", () => {  
  beforeEach(() => {  
    cy.visit("/")  
      .get("main")  
      .injectAxe()  
  })  
  it("Has no detectable accessibility violations on load", () => {  
    // will check for accessibility violations  
    cy.checkA11y()  
  })  
})
```

The Organizers



Donavon West



Harry Wolff



Jenn Creighton



Matthew Gerstman



Yuraima Estevez



© 2019 useReactNYC

Elements Console Sources Network Performance Memory Application

```
<ul class="organizers-ul">
  <li>
    <a href="https://twitter.com/donavon" rel="noopener noreferrer" target="_blank">
      <div>
        
        <p>Donavon West</p>
      </div>
    </a>
  </li>
```

Styles

```
:hov .cls
element.style {
}
(index):1
organizers-ul
li img {
  max-width
```

```
describe('Accessibility tests', () => {  
  // ...  
  
  it('Has alternative text set for all organizer images', () => {  
    cy.get('ul.organizers-ul img')  
      .should(($img) => {  
        // should have 5 img elements  
        expect($img).to.have.length(5)  
  
        // get alt text values for each element  
        const altValues = $img.map((i, el) => {  
          return Cypress.$(el).attr('alt')  
        })  
  
        // there should be 5 alt text values  
        expect(altValues).to.have.lengthOf(5)  
      })  
  })  
})
```

```
it('Has alternative text set for all organizer images', () => {
  cy.get('ul.organizers-ul img')
    .should(($img) => {
      // should have 5 img elements
      expect($img).to.have.length(5)

      // get alt text values for each element
      const altValues = $img.map((i, el) => {
        return Cypress.$(el).attr('alt')
      })

      // there should be 5 alt text values
      expect(altValues).to.have.lengthOf(5)
    })
})

it('Navigates to the "About Us" page and announces page title') {
  // ...
}
```

Tools Testing

Tools Testing

Tools Testing

Thank You!