Engineer

axe

die Engine, die diese Website auf Barrierefreiheit prüft

Was es ist

axe, als quelloffene JavaScript-Bibliothek axe-core veröffentlicht, ist eine Sammlung automatisierter Barrierefreiheitsregeln. Sie läuft innerhalb einer Webseite, untersucht die Browserdarstellung und meldet jedes Element, das eine Regel verletzt. Deque Systems, ein auf Barrierefreiheit spezialisiertes Unternehmen, pflegt sie unter der Mozilla Public License 2.0. Andere Werkzeuge bauen darauf auf: Die Barrierefreiheitskategorie von Chrome Lighthouse gewichtet Prüfungen anhand der axe-Auswirkungsstufen. Auch Playwrights eigener Testleitfaden führt axe über ein kleines Begleitpaket aus.

Die Engine ist ein einzelnes Skript. Ein Testrunner oder eine Browsererweiterung lädt es in die Testseite und ruft `axe.run` auf. Die Engine durchläuft das Dokument samt verschachtelter Frames und Shadow DOM und wendet jede passende Regel an. Weil sie nach der Layoutberechnung liest, kann sie Dinge prüfen, die der Quelltext nicht zeigt: sichtbarer Text, seine Hintergrundfarbe und verborgene Elemente. Deshalb brauchen einige Regeln einen echten Browser. Das Projekt weist darauf hin, dass die Farbkontrastregel in JSDOM, dem in vielen Unittests verwendeten Dokumentmodell, nicht funktioniert.

Jede Regel trägt Tags. Eines ordnet WCAG-Version und Stufe zu, etwa `wcag2a`, `wcag2aa`, `wcag21aa` oder `wcag22aa`. `best-practice` kennzeichnet Regeln ohne verpflichtendes Erfolgskriterium. Andere Tags beziehen sich auf EN 301 549, Section 508 oder die W3C-Konformitätsprüfregeln. Über diese Tags entscheidet ein Team, was ein Lauf prüft.

Das Ergebnis ist nicht einfach bestanden oder durchgefallen, sondern besteht aus vier Listen:

- `violations` : fehlgeschlagene Elemente mit den Auswirkungsstufen minor, moderate, serious oder critical, einer kurzen Reparaturanleitung und einem Link zur längeren Erklärung.
- `passes` : bestandene Elemente, an denen sich erkennen lässt, dass die Regel wirklich lief.
- `incomplete` , in der Dokumentation auch „needs review“: Elemente ohne sichere Entscheidung, die ein Mensch ansehen muss.
- `inapplicable` : Regeln ohne Prüfgegenstand, etwa Videoregeln auf einer Seite ohne Video.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist axe?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Warum es wichtig ist

axe unterstützt die Entscheidung, welche Teile der Barrierefreiheitsprüfung eine Maschine übernimmt und ob ein Fehler die Veröffentlichung stoppen soll. Dafür eignen sich zwei Eigenschaften. Die Ausführung ist günstig und kann jeden Build statt nur ein jährliches Audit prüfen. Außerdem lautet das erklärte Ziel der Hersteller: null Fehlalarme, abgesehen von Programmfehlern. Unsichere Ergebnisse landen in der Prüfliste statt unter Verstößen. Eine ständig falsche Alarmierung wird ausgeschaltet. Eine Prüfung, die nur nachweisbare Fehler meldet, darf einen Build stoppen, ohne Widerstand zu erzeugen.

Die Grenze liegt in der Abdeckung. Deque sagt, axe-core finde durchschnittlich 57 Prozent der WCAG-Probleme automatisch und gebe unklare Ergebnisse zur manuellen Prüfung aus. Die Zahl beschreibt gefundene Probleme, nicht den Anteil der Erfolgskriterien, den eine Maschine beurteilen kann.

Außer Reichweite liegt vor allem das Urteil. axe kann feststellen, ob ein Bild Alternativtext, eine Schaltfläche einen Namen und ein Formularfeld eine Beschriftung besitzt. Es erkennt nicht, ob der Text das Bild erklärt, der Name die

Handlung beschreibt oder die Seite beim Vorlesen in Reihenfolge Sinn ergibt. Dafür braucht es einen Menschen mit Screenreader und Tastatur. Die Zeitersparnis bei mechanischen Prüfungen schafft dafür Raum.

Auch die Tagwahl hat einen Preis. Nur WCAG-Tags liefern ein sauber einer Konformitätsaussage zuordenbares Ergebnis und sonst nichts. Best-Practice-Regeln finden zusätzliche Nutzungsprobleme, etwa Inhalte außerhalb jedes Landmarks oder eine fehlende oberste Überschrift. Dafür entstehen einige Befunde, die kein Konformitätsaudit zitieren würde.

Wie es in der Praxis aussieht

Zwei Seiten zeigen, was der Engine anvertraut wird.

In der Testsuite einer Plattform steht sie neben vier Testschichten. Ein axe-core-Gate verlangt null Verstöße gegen WCAG 2.0 und 2.1 A und AA. Barrierefreiheit wird damit zum Buildfehler statt zum Auditbefund Monate später. Entscheidend ist der Zeitpunkt: Solange die Änderung noch offen ist, repariert die Person den Fehler, die ihn gerade geschrieben hat.

Die Credits dieser Website beschreiben repräsentative Seiten, je eine pro Vorlage in jeder Sprache, geprüft gegen WCAG 2.2 A und AA in beiden Farbschemata. Das geschieht im verbindlichen Gate, nicht in einem Bericht daneben. Beide Schemata erlauben, auch einen nur im dunklen Modus auftretenden Kontrastfehler als Buildfehler zu behandeln. Die Stichprobe setzt gemeinsame Strukturen derselben Vorlage voraus. So konnten im Oktober 2026 insgesamt 38 Seiten für eine Website mit 2.268 Seiten stehen. Automatisierte Regeln decken nicht jede von WCAG verlangte Beurteilung ab. Deshalb führt die Website Aufzeichnungen pro Erfolgskriterium statt eines einzigen Häkchens.

Wo es schiefgeht

Meist scheitert Vertrauen: Ein Lauf prüft weniger als gedacht oder ein Ergebnis wird weiter ausgelegt, als es reicht.

Eine Tagliste lässt Regeln aus. Tags sind nicht kumulativ. `wcag2aa` umfasst nur Regeln für AA-Kriterien aus WCAG 2.0. Ein darauf beschränkter Lauf überspringt daher sämtliche Regeln der Stufe A. Die API-Dokumentation verlangt ausdrücklich sowohl `wcag2a` als auch `wcag2aa`. Ebenso umfasst `wcag21aa` nur in

2.1 ergänzte Kriterien. Ein auf WCAG-Tags begrenzter Lauf führt nie eine `best-practice` -Regel aus. Die Regelbeschreibungen nennen WCAG-2.2-Regeln als standardmäßig deaktiviert. Auch ein Lauf ohne Tags lässt sie aus; ihre ausdrückliche Nennung aktiviert sie. Benennen Sie alle benötigten Tags und halten Sie die Liste neben der von ihr getragenen Aussage:

```
const results = await axe.run(document, {
    runOnly: {
        type: "tag",
        values: ["wcag2a", "wcag2aa", "wcag21a", "wcag21aa", "wcag22aa"],
    },
});
```

Eine Regel in keiner der vier Listen ist überhaupt nicht gelaufen. Lesen Sie die Listen deshalb bei jeder Konfigurationsänderung einmal durch.

Ein sauberer Lauf gilt als barrierefreie Seite. Null Verstöße heißt, dass die ausgeführten Regeln keinen Fehler nachweisen konnten. Über richtigen Alternativtext, sinnvolle Fokusreihenfolge oder den Abschluss einer Aufgabe per Screenreader sagt das nichts. Behandeln Sie den Lauf als Grundlage und planen Sie die Prüfungen, die er nicht kann.

Die Prüfliste wird verworfen. Ein Gate, das nur Verstöße ablehnt, lässt `incomplete` ohne Sichtung durch. Gerade diese untersuchten, unentschiedenen Elemente brauchen am dringendsten einen Menschen. Leiten Sie sie an jemanden weiter, der sie liest, oder lassen Sie auch daran den Build scheitern.

Es wird nur ein Zustand geprüft. Die Engine prüft die Darstellung zum Zeitpunkt ihres Laufs. Ein geschlossenes Menü, ein ungeöffneter Dialog, eine erst nach fehlerhafter Eingabe sichtbare Meldung und das andere Farbschema liegen außerhalb des ersten Seitenzustands. Führen Sie die Seite in jeden wichtigen Zustand und prüfen Sie jeden erneut.

Die Engine läuft dort, wo nichts gezeichnet wird. Ein simuliertes Dokument besitzt weder Layout noch berechnete Farben. Die Farbkontrastregel funktioniert laut Projekt dort nicht. Unittests mit axe finden weiterhin Strukturfehler; Kontrast und alles andere Visuelle gehören in einen echten Browser.

Eine Regel wird für die Veröffentlichung abgeschaltet. Die globale Abschaltung wegen einer einzigen Komponente versteckt dieselben Fehler anderswo, auch auf später geschriebenen Seiten. Nehmen Sie stattdessen nur dieses Element aus und notieren Sie Grund und Aufhebungszeitpunkt.

Weiterführende Lektüre

- axe-core — Quellcode, erklärte Prinzipien und unterstützte Browser.
- Regelbeschreibungen — Auswirkungen und Tags aller Regeln, gruppiert nach WCAG-Version, Best Practice, AAA, experimentell und abgekündigt.
- axe-core API — Tags, `runOnly`, Ein- und Ausschluss von Seitenteilen sowie die vier Ergebnislisten.
- Barrierefreiheit testen — Playwrights Anleitung für axe in Browsertests, einschließlich Prüfungen nach Interaktionen.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist axe?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/axe/