Engineer

axe

den tilgængelighedsmotor dette site revideres med

Hvad det er

axe, udgivet som open source-JavaScript-biblioteket axe-core, er et sæt automatiske tilgængelighedsregler, der kører inde i en webside, undersøger det, browseren har renderet, og rapporterer hvert element, der bryder en af reglerne. Deque Systems vedligeholder det under Mozilla Public License 2.0. Andre værktøjer bygger på det: tilgængelighedskategorien i Chromes Lighthouse vægter sine audits efter axes vurderinger af, hvor hårdt en fejl rammer brugeren, og Playwrights egen vejledning i tilgængelighedstest kører axe gennem en lille tilhørende pakke.

Motoren er ét script. En testkører eller en browserudvidelse indlæser det i den side, der testes, og kalder `axe.run`, og motoren gennemgår dokumentet, også indlejrede frames og shadow DOM, og anvender hver regel, der gælder for det, den møder. Fordi den læser siden, efter at browseren har lagt den ud, kan den tjekke ting, kildekoden alene ikke viser: om en tekst er synlig, hvilken farve den står på, om et element er skjult. Derfor kræver nogle regler en rigtig browser, og projektet skriver ligeud, at dets kontrastregel ikke virker i JSDOM, det simulerede dokument, som mange opsætninger til enhedstest bruger.

Hver regel har tags. Ét af dem siger, hvilken WCAG-version og hvilket niveau reglen hører til, for eksempel `wcag2a`, `wcag2aa`, `wcag21aa` eller `wcag22aa`, eller markerer den som `best-practice`, når intet succeskriterium kræver den. Andre tags knytter regler til EN 301 549, Section 508 eller W3C's regler for konformitetstest. Det er med tags, et team bestemmer, hvad en kørsel tjekker.

Det, der kommer tilbage, er ikke ét samlet bestået eller dumpet, men fire lister:

- `violations` : elementer, der brød en regel, hver med en påvirkning vurderet som minor, moderate, serious eller critical, et kort resumé af rettelsen og et link til en længere forklaring.
- `passes` : elementer, der bestod, og som viser, at en regel faktisk kørte.
- `incomplete` , som dokumentationen også kalder »needs review«: elementer, motoren ikke entydigt kan erklære bestået eller dumpet, og som et menneske skal se på.
- `inapplicable` : regler, der ikke havde noget at teste, for eksempel videoreglerne på en side uden video.

Brug begrebet i en beslutning

Brug »Hvad er axe?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

Har styret full-stack-udviklingen af GIS-kort Kontakt os →

Hvorfor det betyder noget

Den beslutning, axe tjener, er, hvilke dele af tilgængeligheden et team overlader til en maskine, og om en fejl skal stoppe en udgivelse. To egenskaber gør motoren egnet til det. Den er billig at køre, så den kan tjekke hvert build i stedet for én revision om året. Og udviklernes erklærede mål er nul falske positiver, fejl i selve motoren undtaget: alt, hvad motoren er usikker på, havner på listen til gennemsyn og ikke på listen over overtrædelser. Et tjek, der råber ulv, bliver slået fra; et, der kun melder det, det kan bevise, kan få et build til at fejle, uden at nogen bærer nag.

Grænsen er dækningen. Deque oplyser, at axe-core i gennemsnit finder 57 % af WCAG-problemerne automatisk og sender usikre resultater til manuel vurdering. Tallet handler om fundne problemer, ikke om, hvor stor en del af succeskriterierne en maskine kan vurdere.

Det, der ligger uden for rækkevidde, er mest skøn. axe kan bekræfte, at et billede har alternativ tekst, at en knap har et navn, og at et formularfelt har en etiket. Den kan ikke afgøre, om teksten beskriver billedet, om navnet siger, hvad knappen gør, eller om siden giver mening, når den læses højt i rækkefølge. De spørgsmål kræver et menneske med en skærmlæser og et tastatur.

Valget af tags er en afvejning for sig. Kører man kun de WCAG-taggede regler, får man et resultat, der kan stå bag en konformitetserklæring og intet andet. Lægger man best-practice-reglerne til, fanger man problemer, som WCAG ikke nævner, men som brugere stadig møder, for eksempel indhold uden for alle landmærker eller en side uden overskrift på øverste niveau. Prisen er nogle få fejl, som en konformitetsrevision aldrig ville henvise til.

Sådan ser det ud i praksis

To sider på sitet viser, hvad motoren får ansvar for.

I én platforms testsuite står den ved siden af fire lag af test, og en axe-core-gate kræver nul WCAG 2.0- og 2.1-overtrædelser på niveau A og AA, hvilket gør tilgængelighed til en fejl i buildet frem for et revisionsfund måneder senere. Pointen er timingen: en fejl, der får buildet til at fejle, bliver rettet af den, der lige har skrevet den, mens ændringen stadig ligger åben foran vedkommende.

På sitet her beskriver kolofonen, hvordan motoren gennemgår et udvalg af repræsentative sider, én pr. skabelon på hvert sprog, mod WCAG 2.2 A og AA i begge farvetemaer, som en del af porten og ikke som en rapport ved siden af den. At køre i begge farvetemaer betyder, at en kontrastfejl, der kun viser sig i det mørke tema, stadig kan stoppe buildet. At tage én side pr. skabelon bygger på, at sider fra samme skabelon deler struktur, og det var det, der i oktober 2026 lod 38 sider stå for et site på 2.268. Automatiske regler dækker ikke alle vurderinger i WCAG, og derfor fører sitet sin optegnelse pr. succeskriterium i stedet for som ét samlet bestået.

Hvor det går galt

De fleste fejl med axe er fejl i tilliden: en kørsel, der tjekkede mindre, end det så ud til, eller et resultat, der bliver læst som mere, end det siger.

En tagliste udelader regler. Tags lægges ikke oven i hinanden. `wcag2aa` rummer kun reglerne for kriterier på niveau AA i WCAG 2.0, så en kørsel begrænset til det tag springer alle regler på niveau A over, og API-dokumentationen siger direkte, at både `wcag2a` og `wcag2aa` skal med. På samme måde rummer `wcag21aa` kun regler for de kriterier, 2.1 tilføjede, og en kørsel begrænset til WCAG-tags udfører aldrig en `best-practice` -regel. Regelbeskrivelserne angiver WCAG 2.2-

reglerne som slået fra som standard, så en kørsel uden tags springer dem over; nævner man deres tag, slås de til. Nævn hvert tag, kørslen skal dække, og hold listen ved siden af den påstand, den understøtter:

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

En regel, der står på ingen af de fire lister, kørte slet ikke, så læs listerne igennem, hver gang konfigurationen ændres.

En ren kørsel bliver læst som en tilgængelig side. Nul overtrædelser betyder, at de regler, der kørte, ikke kunne bevise en eneste fejl. Det siger intet om, hvorvidt den alternative tekst er rigtig, om fokus flytter sig i en fornuftig rækkefølge, eller om en skærmlæserbruger kan gennemføre opgaven. Behandl kørslen som gulvet, og planlæg de tjek, den ikke selv kan lave.

Listen til gennemsyn bliver smidt væk. En gate, der kun fejler på overtrædelser, lukker `incomplete`-resultaterne igennem, uden at nogen ser dem. Det er de elementer, motoren har kigget på og ladet stå uafgjort, og derfor er det dem, der mest har brug for et menneske. Send dem til en, der vil læse dem, eller lad buildet fejle på dem også.

Siden bliver testet i én tilstand. Motoren tjekker det, der er renderet i det øjeblik, den kører. En lukket menu, en dialog, der aldrig er blevet åbnet, en fejlmeddelelse, der først viser sig efter en forkert indsendelse, og det andet farvetema ligger alle uden for en kørsel mod siden, som den ser ud ved første indlæsning. Før siden ind i hver tilstand, der betyder noget, og kør motoren igen i hver af dem.

Motoren kører, hvor intet bliver tegnet. Et simuleret dokument har hverken layout eller beregnede farver, og projektet nævner kontrast som en regel, der ikke virker dér. Enhedstest med axe fanger stadig strukturelle problemer; kontrast og alt andet visuelt hører hjemme i en rigtig browser.

En regel bliver slået fra for at få en udgivelse ud. At deaktivere en regel for hele kørslen, fordi én komponent udløser den, skjuler reglens fejl alle andre steder, også i sider, der skrives senere. Udelad i stedet det ene element, med en note om, hvorfor og hvornår undtagelsen skal fjernes.

Læs mere

- axe-core — motorens kildekode, dens erklærede principper og listen over understøttede browsere.
- Rule Descriptions — hver regel med påvirkning og tags, grupperet efter WCAG-version, best practice, AAA, eksperimentelle og forældede regler.
- axe-core API — tags, indstillingen `runOnly`, hvordan dele af en side tages med eller udelades, og formen på de fire resultatlister.
- Accessibility testing — Playwrights vejledning i at køre axe i browsertest, også efter at man har interageret med siden.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er axe?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/axe/