Engineer

axe

motoren nettstedets universelle utforming revideres med

Hva det er

axe, publisert som JavaScript-biblioteket axe-core med åpen kildekode, er et sett automatiserte regler for universell utforming som kjører inne i en nettside, undersøker det nettleseren har gjengitt, og melder fra om hvert element som bryter én av reglene. Deque Systems, et selskap som arbeider med universell utforming, vedlikeholder det under Mozilla Public License 2.0. Andre verktøy bygger på det: kategorien for tilgjengelighet i Lighthouse fra Chrome vekter revisjonene sine etter alvorlighetsgradene til axe, og Playwrights egen veiledning i testing av tilgjengelighet kjører axe gjennom en liten tilleggspakke.

Motoren er ett enkelt skript. En testkjører eller en nettleserutvidelse laster det inn i siden som testes, og kaller `axe.run`, og motoren går gjennom dokumentet, også nøstede rammer og shadow DOM, og bruker hver regel der betingelsene stemmer. Fordi den leser siden etter at nettleseren har lagt den ut, kan den sjekke ting kildekoden alene ikke viser: om teksten er synlig, hvilken farge den står på, om et element er skjult. Det er også grunnen til at noen regler trenger en ekte nettleser. Prosjektet sier at regelen for fargekontrast ikke virker i JSDOM, det simulerte dokumentet mange oppsett for enhetstester bruker.

Hver regel har merkelapper (tags). Én av dem sier hvilken WCAG-versjon og hvilket nivå regelen hører til, f.eks. `wcag2a`, `wcag2aa`, `wcag21aa` eller `wcag22aa`, eller merker den `best-practice` når ingen suksesskriterier krever den. Andre merkelapper knytter regler til EN 301 549, Section 508 eller W3Cs testregler for samsvar. Merkelappene er måten et team bestemmer hva en kjøring sjekker på.

Det som kommer tilbake, er ikke ett enkelt bestått eller ikke bestått, men fire lister:

- `violations` : elementer som brøt en regel, hver med en alvorlighetsgrad satt til minor, moderate, serious eller critical, et kort sammendrag av hvordan det rettes, og en lenke til en lengre forklaring.
- `passes` : elementer som besto, og det er slik et team kan se at en regel faktisk ble kjørt.
- `incomplete` , som dokumentasjonen også kaller «needs review»: elementer motoren verken kunne godkjenne eller underkjenne med visshet, og som en person må se på.
- `inapplicable` : regler som ikke hadde noe å teste, som reglene for video på en side uten video.

Bruk begrepet i en beslutning

Bruk «Hva er axe?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Beslutningen axe hjelper med, er hvilke deler av universell utforming et team overlater til en maskin, og om et brudd skal stoppe en utgivelse. To egenskaper gjør den egnet til den jobben. Den er billig å kjøre, så den kan sjekke hvert bygg i stedet for én revisjon i året. Og målet utviklerne har uttalt, er null falske positive, bortsett fra programvarefeil: alt motoren er usikker på, havner på listen til gjennomgang i stedet for på listen over brudd. En sjekk som roper ulv, blir slått av; en som bare melder det den kan bevise, kan stoppe et bygg uten at noen blir irritert.

Grensen er dekningen. Deque sier at axe-core i gjennomsnitt finner 57 % av WCAG-problemene automatisk og sender usikre resultater til manuell vurdering. Tallet gjelder problemer som blir funnet, ikke andelen suksesskriterier en maskin kan vurdere.

Det som ligger utenfor rekkevidde, er for det meste skjønn. axe kan bekrefte at et bilde har alternativ tekst, at en skjermknapp har et navn og at et skjemafelt har en etikett. Den kan ikke si om teksten beskriver bildet, om navnet sier hva

knappen gjør, eller om siden gir mening når den leses høyt i rekkefølge. De spørsmålene trenger en person med en skjermleser og et tastatur, og tiden en maskin sparer på de mekaniske sjekkene, er tid den personen kan bruke på dem.

Valget av merkelapper er en avveining i seg selv. Å kjøre bare reglene som er merket for WCAG, gir et resultat som passer rett inn i en samsvarserklæring, og ikke i noe annet. Å legge til reglene for beste praksis fanger opp problemer WCAG ikke nevner, men som brukerne likevel møter, som innhold som ligger utenfor hvert landemerke, eller en side uten overskrift på øverste nivå. Prisen er noen få brudd som ingen samsvarsrevisjon ville vist til.

Slik ser det ut i praksis

To sider på dette nettstedet viser hva motoren blir betrodd.

I testsuiten til én plattform står den ved siden av fire lag med tester, og en kvalitetsport med axe-core krever null brudd på WCAG 2.0 og 2.1 på nivå A og AA, noe som gjør universell utforming til et bygg som feiler, i stedet for et funn i en revisjon flere måneder senere. Poenget er tidspunktet: en feil som stopper bygget, blir rettet av personen som nettopp skrev den, mens endringen fortsatt er åpen.

På dette nettstedet beskriver kolofonen hvordan motoren reviderer et utvalg representative sider, én per mal på hvert språk, mot WCAG 2.2 A og AA i begge fargeskjemaene, som en del av kontrollen og ikke som en rapport ved siden av den. At den kjører i begge fargeskjemaene, betyr at et kontrastbrudd som bare viser seg i det mørke skjemaet, fortsatt kan stoppe bygget. Å teste én side per mal bygger på at sider laget fra samme mal deler struktur, og det var det som i oktober 2026 lot 38 sider stå for et nettsted på 2 268. Automatiske regler dekker ikke alle vurderingene WCAG krever, og derfor fører nettstedet oversikten sin per suksesskriterium i stedet for som en samlet karakter.

Hvor det går galt

De fleste feil med axe er feil i tillit: en kjøring som sjekket mindre enn det så ut som, eller et resultat som blir lest som mer enn det sier.

En liste med merkelapper utelater regler. Merkelappene er ikke kumulative. `wcag2aa` inneholder bare reglene for kriterier på nivå AA i WCAG 2.0, så en kjøring som er begrenset til den, hopper over hver regel på nivå A, og API-dokumentasjonen sier rett ut at både `wcag2a` og `wcag2aa` trengs. På samme måte inneholder `wcag21aa` bare regler for kriterier som kom med 2.1, og en kjøring som er begrenset til WCAG-merkelapper, kjører aldri en `best-practice`-regel. Regelbeskrivelsene oppgir reglene for WCAG 2.2 som slått av som standard, så en kjøring uten merkelapper utelater dem; når du nevner merkelappen deres, slås de på. Nevn hver merkelapp kjøringen skal dekke, og hold listen ved siden av påstanden den støtter:

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

En regel som ikke står i noen av de fire listene, ble ikke kjørt i det hele tatt, så les gjennom listene én gang hver gang konfigurasjonen endres.

En kjøring uten brudd blir lest som en tilgjengelig side. Null brudd betyr at reglene som ble kjørt, ikke kunne bevise noen feil. Det sier ingenting om hvorvidt den alternative teksten er riktig, om fokus flytter seg i en fornuftig rekkefølge, eller om noen som bruker en skjermleser, kan fullføre oppgaven. Behandle kjøringen som et minstemål, og sett av tid til sjekkene den ikke kan gjøre.

Listen til gjennomgang blir droppet. En kvalitetsport som bare feiler på brudd, slipper `incomplete`-resultatene gjennom uten at noen ser dem. Det er elementene motoren så på og lot stå uavgjort, og det gjør dem til tilfellene som mest trenger en person. Send dem til noen som kommer til å lese dem, eller la bygget feile på dem også.

Siden testes i én tilstand. Motoren sjekker det som er gjengitt i øyeblikket den kjører. En lukket meny, en dialog som ikke er åpnet, en feilmelding som bare dukker opp etter en ugyldig innsending, og det andre fargeskjemaet ligger alle utenfor en kjøring mot siden slik den først lastes inn. Før siden inn i hver tilstand som betyr noe, og kjør motoren på nytt i hver av dem.

Motoren kjører der ingenting tegnes. Et simulert dokument har verken layout eller beregnede farger, og prosjektet oppgir fargekontrast som en regel som ikke virker der. Enhetstester med axe fanger fortsatt opp strukturelle problemer; kontrast og alt annet som er visuelt, hører hjemme i en ekte nettleser.

En regel slås av for å få ut en utgivelse. Å slå av en regel for hele kjøringen fordi én komponent snubler i den, skjuler bruddene på den regelen alle andre steder, også på sider som skrives senere. Utelat heller det ene elementet, med en merknad om hvorfor og når unntaket skal fjernes.

Les mer

- axe-core — kildekoden til motoren, prinsippene prosjektet har uttalt, og nettleserne som støttes.
- Rule Descriptions — hver regel med alvorlighetsgrad og merkelapper, gruppert etter WCAG-versjon, beste praksis, AAA, eksperimentelle og utfasede regler.
- axe-core API — merkelappene, valget `runOnly`, hvordan du tar med eller utelater deler av en side, og formen på de fire resultatlistene.
- Accessibility testing — Playwrights veiledning i å kjøre axe i nettlesertester, også å skanne en side etter at du har samhandlet med den.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er axe?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/axe/