Engineer

axe

tillgänglighetsmotorn som den här webbplatsen granskas med

Vad det är

Regelmotorn axe, som ges ut som JavaScript-biblioteket axe-core med öppen källkod, är en uppsättning automatiska regler för tillgänglighet som körs inuti en webbsida, granskar det som webbläsaren har renderat och rapporterar varje element som bryter mot någon av reglerna. Deque Systems, ett företag som arbetar med tillgänglighet, underhåller den under Mozilla Public License 2.0. Andra verktyg bygger på den: tillgänglighetskategorin i Chromes Lighthouse viktar sina granskningar efter de allvarlighetsgrader som axe sätter, och Playwrights egen guide till tillgänglighetstester kör axe via ett litet tilläggspaket.

Motorn är ett enda skript. En testkörare eller ett webbläsartillägg läser in det i sidan som testas och anropar `axe.run`, och motorn går igenom dokumentet, inklusive inbäddade ramar och shadow DOM, och tillämpar varje regel vars villkor stämmer. Eftersom den läser sidan efter att webbläsaren har lagt ut den kan den kontrollera sådant som källkoden ensam inte visar: om texten syns, vilken färg den står mot, om ett element är dolt. Det är också därför vissa regler behöver en riktig webbläsare. Projektet anger att dess regel för färgkontrast inte fungerar i JSDOM, det simulerade dokument som många uppsättningar för enhetstester använder.

Varje regel bär taggar. En av dem anger vilken version och nivå av WCAG som regeln hör till, t.ex. `wcag2a`, `wcag2aa`, `wcag21aa` eller `wcag22aa`, eller märker den `best-practice` när inget framgångskriterium kräver den. Andra taggar knyter regler till EN 301 549, Section 508 eller W3C:s regler för konformitetstestning. Det är med taggarna som ett team bestämmer vad en körning kontrollerar.

Det som kommer tillbaka är inte ett enda godkänt eller underkänt utan fyra listor:

- `violations` : element som föll på en regel, vart och ett med en allvarlighetsgrad, minor, moderate, serious eller critical, en kort sammanfattning av hur det åtgärdas och en länk till en längre förklaring.
- `passes` : element som klarade regeln, vilket är hur ett team kan se att en regel faktiskt kördes.
- `incomplete` , som dokumentationen också kallar ”needs review”: element som motorn varken kunde godkänna eller underkänna med säkerhet, och som en människa måste titta på.
- `inapplicable` : regler som inte hade något att testa, t.ex. reglerna för video på en sida utan video.

Använd begreppet i ett beslut

Använd ”Vad är axe?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Styrde full stack-utvecklingen av GIS-kartor — kartutveckling i full stack, från databas till webbläsare

Styrde full stack-utvecklingen av GIS-kartor Hör av er →

Varför det spelar roll

Beslutet som axe tjänar är vilka delar av webbtillgängligheten ett team lämnar över till en maskin, och om ett fynd ska stoppa en release. Två egenskaper gör motorn lämpad för det. Den är billig att köra, så den kan kontrollera varje bygge i stället för en granskning om året. Och upphovspersonernas uttalade mål är noll falska positiva, buggar undantagna: allt som motorn är osäker på hamnar i granskningslistan i stället för i listan över överträdelser. En kontroll som ropar varg blir avstängd; en som bara rapporterar det den kan bevisa kan stoppa ett bygge utan att någon tar illa upp.

Begränsningen är täckningen. Deque uppger att axe-core i genomsnitt hittar 57 % av WCAG-problemen automatiskt och lämnar osäkra resultat till manuell granskning. Siffran gäller hittade problem, inte andelen framgångskriterier som en maskin kan bedöma.

Det som ligger utom räckhåll är mest omdöme. Motorn kan bekräfta att en bild har en alternativtext, att en knapp har ett namn och att ett formulärfält har en etikett. Den kan inte avgöra om texten beskriver bilden, om namnet säger vad knappen gör eller om sidan hänger ihop när den läses upp i ordning. De frågorna

kräver en människa med en skärmläsare och ett tangentbord, och den tid som en maskin sparar på de mekaniska kontrollerna är tid som den människan kan lägga på dem.

Valet av taggar är en avvägning i sig. Att bara köra de regler som är taggade för WCAG ger ett resultat som svarar rent mot ett påstående om konformitet, och mot inget annat. Att lägga till reglerna för bästa praxis fångar problem som WCAG inte namnger men som användare ändå stöter på, t.ex. innehåll som ligger utanför varje landmärke eller en sida utan rubrik på översta nivån. Kostnaden är några fynd som ingen granskning av konformitet skulle hänvisa till.

Så ser det ut i praktiken

Två sidor på den här webbplatsen visar vad motorn anförtros.

I en plattforms testsvit står den bredvid fyra lager av tester, och en axe-core-grind kräver noll överträdelser av WCAG 2.0 och 2.1 på nivå A och AA, vilket gör tillgänglighet till ett byggfel i stället för ett granskningsfynd månader senare. Poängen är tidpunkten: en defekt som får bygget att fallera rättas av den som just skrev den, medan ändringen fortfarande är öppen.

På den här webbplatsen beskriver kolofonen hur motorn granskar ett urval representativa sidor, en per mall på varje språk, mot WCAG 2.2 A och AA i båda färgschemana, som en del av kvalitetsgrinden och inte som en rapport bredvid den. Att köra i båda färgschemana innebär att ett kontrastproblem som bara visar sig i det mörka schemat ändå kan stoppa bygget. Urvalet med en sida per mall bygger på att sidor som byggs från samma mall delar struktur, och det var det som i oktober 2026 lät 38 sidor stå för en webbplats med 2 268. Automatiska regler täcker inte alla bedömningar i WCAG, och därför för webbplatsen sin redovisning per framgångskriterium i stället för som ett samlat godkänt.

Var det går fel

De flesta misslyckanden med axe är förtroendefel: en körning som kontrollerade mindre än det verkade, eller ett resultat som lästes som mer än det säger.

En tagglista utelämnar regler. Taggar är inte kumulativa. `wcag2aa` innehåller bara reglerna för kriterier på nivå AA i WCAG 2.0, så en körning som begränsas till den hoppar över varje regel på nivå A, och API-dokumentationen säger rakt ut att

både `wcag2a` och `wcag2aa` behövs. På samma sätt innehåller `wcag21aa` bara regler för kriterier som 2.1 lade till, och en körning som begränsas till WCAG-taggar kör aldrig någon regel märkt `best-practice` . Regelbeskrivningarna listar reglerna för WCAG 2.2 som avstängda som standard, så en körning utan taggar utelämnar dem; att namnge deras tagg slår på dem. Namnge varje tagg som körningen ska täcka, och håll listan bredvid det påstående den stöder:

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

En regel som inte finns i någon av de fyra listorna kördes inte alls, så läs igenom listorna en gång varje gång konfigurationen ändras.

En felfri körning läses som en tillgänglig sida. Noll överträdelser betyder att de regler som kördes inte kunde bevisa något fel. Det säger ingenting om huruvida alternativtexten stämmer, om fokus flyttas i en rimlig ordning eller om någon som använder en skärmläsare kan slutföra uppgiften. Behandla körningen som golvet, och planera in de kontroller den inte kan göra.

Granskningslistan tappas bort. En grind som bara fäller på överträdelser släpper igenom resultaten i `incomplete` utan att någon ser dem. Det är de element som motorn tittade på och lämnade obestämda, vilket gör dem till de fall som mest behöver en människa. Skicka dem till någon som läser dem, eller låt bygget fallera på dem också.

Sidan testas i ett enda tillstånd. Motorn kontrollerar det som är renderat i det ögonblick den körs. En stängd meny, en dialog som inte har öppnats, ett felmeddelande som bara visas efter en felaktig inskickning och det andra färgschemat ligger alla utanför en körning mot sidan så som den ser ut när den först laddas. Driv sidan in i varje tillstånd som spelar roll, och kör motorn igen i vart och ett.

Motorn körs där ingenting ritas. Ett simulerat dokument har ingen layout och ingen beräknad färg, och projektet listar färgkontrast som en regel som inte fungerar där. Enhetstester med axe fångar fortfarande strukturella problem;

kontrast och allt annat visuellt hör hemma i en riktig webbläsare.

En regel stängs av för att få ut en release. Att inaktivera en regel för hela körningen för att en komponent löser ut den döljer regelns fynd överallt annars, även på sidor som skrivs senare. Undanta i stället det enda elementet, med en kommentar om varför och när undantaget ska tas bort.

Läs mer

- axe-core — motorns källkod, dess uttalade principer och listan över webbläsare som stöds.
- Rule Descriptions — varje regel med dess allvarlighetsgrad och taggar, grupperade efter version av WCAG, bästa praxis, AAA, experimentella och utfasade.
- axe-core API — taggarna, alternativet `runOnly`, hur delar av en sida tas med eller utesluts och formen på de fyra resultatlistorna.
- Accessibility testing — Playwrights guide till att köra axe inuti webbläsartester, bl.a. hur en sida skannas efter att man har interagerat med den.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är axe?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/axe/