Engineer

Webbrowser

das Programm, mit dem Menschen Webseiten öffnen

Was es ist

Ein Webbrowser ist die Software, die eine Adresse entgegennimmt, die Antwort des Servers abruft und daraus eine Seite macht, die ein Mensch lesen, scrollen und anklicken kann. Im Alltag heißt er einfach Browser. Chrome, Safari, Firefox und Edge sind die Namen auf den Symbolen. Dahinter stecken viel weniger Programme, als die Namen vermuten lassen.

Der HTML-Standard spricht selten von „Browsern“. Sein Begriff lautet *User Agent*: die Software, die im Auftrag des Lesers handelt. Der größte Teil des Standards zählt auf, was ein User Agent mit einer Seite tun muss. Drei dieser Aufgaben erklären den größten Teil dessen, wofür ein Browser da ist.

Er parst. Der Standard legt Parsingregeln für HTML-Dokumente fest, „ob sie syntaktisch korrekt sind oder nicht“. Ein fehlendes End-Tag oder ein falsch platziertes Element hält die Seite deshalb nicht auf. Jeder standardkonforme Browser repariert dasselbe fehlerhafte Markup zum selben Elementbaum. Deshalb erscheint auch eine nachlässig geschriebene Seite, und ihre Fehler können jahrelang unbemerkt bleiben.

Er betreibt eine Ereignisschleife. Klicks, Tastendrücke, Timer, Netzwerkantworten, Skripte und das Zeichnen des nächsten Bildes reihen sich ein und kommen nacheinander an die Reihe. Ein Skript, das die Schleife zwei Sekunden lang belegt, hält damit auch Klicks, Texteingabe und die nächste Darstellung auf. Genau das meint ein Mensch, wenn er sagt, eine Seite hänge.

Er trennt Websites voneinander. Der Standard definiert einen Ursprung als Schema, Host und Port und eine Site als Schema plus registrierbaren Teil der Domain. Greift ein Skript eines Ursprungs in das Dokument eines anderen, erhält

es einen Sicherheitsfehler statt des Inhalts. Der größte Teil dessen, was einen Banktab vor einem Nachrichtentab schützt, ist diese Regel, die der Browser bei jedem Zugriff durchsetzt.

Der Teil, der diese Arbeit erledigt, heißt *Engine*. Die meisten heute verwendeten Browser bauen auf einer von drei Engines auf: Blink in Chrome, Edge und vielen weiteren; WebKit in Safari; Gecko in Firefox. Ein neuer Browsername bedeutet meist eine neue Hülle um eine dieser Engines, mit eigener Adressleiste, Einstellungen und Synchronisierung, aber denselben Regeln zum Lesen und Zeichnen einer Seite.

Zwei benachbarte Dinge werden damit verwechselt. Eine Suchmaschine ist eine Website, die der Browser öffnet, wenn die Adressleiste Wörter statt einer Adresse erhält. Eine *Webview* ist eine in eine andere App eingebettete Browserengine, bei der Adressleiste und die meisten Bedienelemente fehlen. Testwerkzeuge ergänzen den *Headless*-Browser. Er zeichnet Seiten, die niemand ansieht, damit ein Skript sie prüfen kann.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Webbrowser?“ 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

Ein Server kann genau die richtigen Bytes senden, und der Leser kann trotzdem eine unlesbare Seite erhalten. Was ein Mensch sieht, ist das, was sein Browser aus diesen Bytes gemacht hat, ergänzt um seine eigenen Einstellungen. Im Browser wird die Arbeit beurteilt.

Getestet werden Engines, keine Marken. Ein Testplan mit vier Browsernamen kann zwei Engines jeweils doppelt abdecken und die dritte auslassen. Eine brauchbare Liste nennt die Engines und die Plattformen, auf denen sich dieselbe

Engine anders verhält, etwa ein Telefon mit kleinem Bildschirm und Touchbedienung oder einen Druckdialog.

Die Einstellungen des Lesers gehören zur Seite. Ein Browser bringt ein bevorzugtes Farbschema, den Wunsch nach reduzierter Bewegung, eine Zoomstufe, eine Standardtextgröße und manchmal eine Übersetzungsschicht mit. Eine Seite, die all das ignoriert, ist für die eine Konfiguration einer einzigen Person entworfen, meist derjenigen, die sie gebaut hat.

Funktionen werden abgefragt, nicht vorausgesetzt. Engines führen neues CSS und neue Schnittstellen zu unterschiedlichen Zeitpunkten ein. Eine Seite, die etwas braucht, das nur manche Engines beherrschen, kann direkt nachfragen: CSS bietet `@supports`, und ein Skript kann vor der Verwendung prüfen, ob eine Eigenschaft vorhanden ist. Die User-Agent-Zeichenkette ist ein schlechter Ersatz. Browser schreiben sie seit Langem teilweise so, dass sie einander ähneln, damit älterer Code zur Browsererkennung sie durchlässt.

Regeln im Browser sind eine Hilfestellung. Formularvalidierung, ein deaktiviertes Bedienelement oder ein verborgener Link laufen in einem Programm, das dem Leser gehört und das er verändern kann. Sie erleichtern ehrlichen Nutzern die Bedienung und halten niemanden auf, der die Entwicklerwerkzeuge öffnet oder die Seite ganz überspringt. Alles, was Geld, Daten oder Zugriff schützt, wird deshalb auf dem Server erneut geprüft.

Wie es in der Praxis aussieht

Diese Website unterstreicht ihre Links mit Skip-Ink, der Einstellung, die die Linie dort unterbricht, wo sie eine Unterlänge kreuzen würde. In Chromium sahen die Unterbrechungen richtig aus. Bei achtfacher Vergrößerung in einem echten Safari gemessen waren sie ungefähr sechsmal schmaler. Ursache war ein WebKit-Fehler: Ein einziger Schriftschnitt mit deklariertem nicht lateinischem Zeichenbereich beeinträchtigte Skip-Ink für die ganze Schriftfamilie. Die Korrektur war eine umbenannte Familie in der Schriftliste. Der Bericht über die Safari-Unterstreichung und das ursächliche Schrift-Subset endet mit der Lehre, die dieser Artikel an den Anfang stellen würde: Eine Skip-Ink-Messung gilt nur für die Engine, die sie erzeugt hat. Tabelle, Diagnose und erste Korrektur stammten sämtlich aus Chromium. Der Screenshot, der den Fall erneut aufrollte, kam von einem Menschen, der die Seite betrachtete.

Beim Drucken gehen die Engines ein zweites Mal auseinander. Das 637 Zeilen lange Druckstylesheet dokumentiert vier Verhaltensweisen von Engines als Fallen, jede mit einem echten Fehler verknüpft. Eine Engine bemisst Viewporteinheiten nach dem gedruckten Seitenrahmen, eine andere nach dem Browserfenster. In der zweiten ging beim Druck aus einem breiten Fenster deshalb ein Drittel jeder Zeile verloren. Fest positionierte Elemente erscheinen auf jedem Blatt. Für das standardmäßig dunkle Design eingefärbte Beschriftungen wurden bei Lesern mit dunkler Browserpräferenz weiß auf weiß gedruckt. Eine Flex- oder Gridbox, die das Seitenlayout beim Drucken nicht teilen kann, erzeugte ein leeres erstes Blatt. Die schützende Prüfung durchläuft sechs Kombinationen aus Papierformat und Ausrichtung mit beiden Farbschemapräferenzen und zählt die tatsächlichen Seiten. Eine Engine ignoriert weiterhin die Kontrolle von Schusterjungen und Hurenkindern.

Der dritte Fall wurde von Lesern entdeckt, nicht durch einen Test. Als die Website als Tor-Onion-Spiegel unter einer lesbaren Adresse veröffentlicht wurde, verwendeten ihre ersten Leser eine andere Browserengine. Diese implementiert die scrollgesteuerte Animation nicht, auf die die Fußzeile angewiesen war. Bei allen wurde das Kolophon über die Startseite gezeichnet. Der Fehler bestand auch auf der gewöhnlichen Website. Erst ein Publikum mit einer anderen Engine machte ihn sichtbar.

Wo es schiefgeht

- **Eine Seite wird im Browser auf dem Entwicklerschreibtisch geprüft und für fertig erklärt.** Jede Eigenheit dieser einen Engine wird zur versteckten Annahme. Abhilfe schafft eine kurze Matrix mit einem Browser pro Engine sowie einem Telefon. In jedem davon wird gemessen, bevor eine Korrektur als fertig gilt.
- **Code liest den Browsernamen und errät dessen Fähigkeiten.** Die Vermutung ist bei jedem Browser falsch, der sich den Namen eines anderen leiht, und veraltet mit jeder Veröffentlichung. Die Abfrage der Funktion überlässt die Entscheidung der Engine. Die Regel außerhalb des Blocks ist das, was jede andere Engine erhält. Deshalb sollte sie den sicheren Zustand beschreiben:

```
.footer {
    position: static;
}

@supports (animation-timeline: scroll()) {
    .footer {
```

```
        animation: settle linear both;
        animation-timeline: scroll();
    }
}
```

- **Ungestaltete Elemente erben die Voreinstellungen des Browsers.** Überschriften, Abstände und besonders Formularelemente bringen das eingebaute Stylesheet der jeweiligen Engine mit. Diese Stylesheets unterscheiden sich. Ein fertig wirkender Entwurf kann auf Entscheidungen einer Engine beruhen, ohne dass jemand sie bewusst getroffen hat.
- **Eine nur auf der Seite durchgesetzte Regel gilt als verbindlich durchgesetzt.** Eine per Skript ausgegraute Schaltfläche, ein auf Zahlen beschränktes Feld, ein für Menschen ohne Tarif verborgener Link: Das gilt jeweils nur für Leser, die die Seite so lassen, wie sie gesendet wurde. Der Server wiederholt jede wichtige Prüfung.
- **Links öffnen sich innerhalb einer fremden App.** Ein in einem Chat oder sozialen Medium angetippter Link landet oft in einem In-App-Browser oder einer Webview, häufig mit eigenem Speicher. Wer im Hauptbrowser angemeldet ist, kommt deshalb als Unbekannter an. Abläufe, die an eine andere App übergeben oder eine bestehende Sitzung voraussetzen, müssen auch von diesen Einstiegspunkten aus getestet werden.

Weiterführende Lektüre

- NIST-CSRC-Glossar: Webbrowser — zwei kurze Definitionen aus NIST-Veröffentlichungen, darunter die schlichte „Clientsoftware zum Anzeigen von Webinhalten“.
- HTML-Standard: HTML-Dokumente parsen — die Regeln, nach denen jeder Browser aus fehlerhaftem oder korrektem Markup denselben Baum erzeugt.
- HTML-Standard: Ereignisschleifen — wie Eingaben, Skripte, Netzwerkantworten und Darstellung in einer Schleife nacheinander an die Reihe kommen.
- HTML-Standard: Webseiten laden — Ursprünge, Sites und die Regeln, die eine Seite daran hindern, eine andere zu lesen.
- WebKit Bugzilla: Fehler 255159 — der eigene Fehlerbericht der Engine zum oben beschriebenen Skip-Ink-Problem.

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 ein Webbrowser?“ 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/browser/