Engineer

Браузер

програма, якою людина відкриває вебсторінки

Що це таке

Браузер — це програма, яка бере адресу, отримує відповідь сервера і складає з неї сторінку, яку можна читати, гортати й натискати; повна назва — веббраузер, але так його мало хто називає. Chrome, Safari, Firefox і Edge — це назви на значках. Під ними працює значно менше окремих програм, ніж можна подумати з цих назв.

Стандарт HTML рідко вживає слово «браузер». Він говорить про *user agent* — програму, яка діє від імені читача, — і більша частина стандарту описує, що така програма мусить робити зі сторінкою. Три з цих обов'язків пояснюють майже все, для чого браузер існує.

Перший — розбір розмітки. Стандарт задає правила розбору HTML-документів незалежно від того, чи вони синтаксично правильні. Тому пропущений закривальний тег або елемент не на своєму місці не зупиняють сторінку: кожен браузер, що дотримується стандарту, лагодить ту саму зламану розмітку в те саме дерево елементів. Ось чому недбала сторінка все одно відкривається — і чому помилка в ній може роками лишатися непоміченою.

Другий — цикл подій, англійською *event loop*. Натискання, клавіші, таймери, відповіді мережі, скрипти й малювання наступного кадру стають у чергу й виконуються по черзі. Скрипт, що тримає цикл дві секунди, затримує разом із собою і натискання, і введення тексту, і наступне оновлення екрана. Саме це люди мають на увазі, коли кажуть, що сторінка «зависла».

Третій — розмежування сайтів. Стандарт визначає походження, *origin*, як поєднання схеми, хоста й порту, а сайт — як схему разом із тією частиною домену, яку реєструють. Скрипт з одного походження, що спробує зазирнути

в документ іншого, отримає помилку безпеки замість вмісту. Здебільшого саме це правило, яке браузер застосовує під час кожного звернення, тримає вкладку з банком окремо від вкладки з новинами.

Усю цю роботу виконує *рушій* браузера. Більшість браузерів, якими сьогодні користуються, побудовано на одному з трьох: Blink у Chrome, Edge та багатьох інших, WebKit у Safari і Gecko у Firefox. Нова назва браузера здебільшого означає нову оболонку навколо одного з них, зі своїм адресним рядком і налаштуваннями, але з тими самими правилами читання й малювання сторінки.

Із браузером часто плутають двох сусідів. Пошукова система — це вебсайт, який браузер відкриває, коли в адресний рядок вводять слова замість адреси. Webview — це рушій браузера, вбудований в інший застосунок, без адресного рядка й більшості кнопок керування. Інструменти тестування також використовують безголовий (headless) браузер: він відмальовує сторінки без видимого вікна, щоб скрипт міг їх перевірити.

Застосуйте поняття до рішення

Застосуйте «Що таке браузер?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Інтерфейси, що швидко завантажуються, працюють на телефоні й поводяться однаково в усіх браузерах ваших клієнтів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала full-stack-розробкою GIS-карт — full-stack-картографія — від бази даних до браузера

Керувала full-stack-розробкою GIS-карт Зв'язатися з нами →

Чому це важливо

Сервер може надіслати саме ті байти, які треба, а читач однаково отримає нечитабельну сторінку: людина бачить те, що з цих байтів склав її браузер, із її власними налаштуваннями поверх. Роботу зрештою оцінюють саме в браузері.

Тестують рушії, а не марки. План тестування з чотирма назвами браузерів може двічі перевіряти два рушії й зовсім оминути третій. Корисний список називає рушії та ситуації, у яких той самий рушій поводиться інакше, — скажімо, телефон із сенсорним екраном чи діалог друку.

Налаштування читача — частина сторінки. Браузер несе бажану колірну схему, запит на зменшення руху, масштаб, типовий розмір шрифту, а часом і вбудований переклад. Сторінка, яка все це ігнорує, зроблена для однієї конфігурації однієї людини, найчастіше тієї, що її створила.

Можливість перевіряють, а не вгадують. Рушії отримують нові можливості CSS і нові інтерфейси в різний час. Сторінка, якій потрібне те, що вміють лише деякі рушії, може спитати рушій напряму: у CSS є `@supports`, а скрипт може перевірити властивість, перш ніж її використати. Рядок user agent — погана заміна, бо браузери давно пишуть його так, щоб скидатися один на одного й проходити повз старий код, який шукав назви.

Правила в браузері — це ввічливість. Перевірка форм, вимкнена кнопка чи приховане посилання працюють у програмі, якою володіє читач і яку він може змінити. Вони полегшують чесний шлях, але не зупиняють того, хто відкриє інструменти розробника або взагалі обійде сторінку. Тому все, що захищає гроші, дані чи доступ, сервер перевіряє ще раз.

Як це виглядає на практиці

Цей сайт підкреслює посилання з увімкненим skip-ink — налаштуванням, яке розриває лінію там, де вона перетнула б нижній виносний елемент літери. У Chromium розриви мали правильний вигляд. Виміряні в справжньому Safari у восьмикратному масштабі, вони виявилися приблизно вшестеро вужчими, а причиною стала вада WebKit: одне накреслення шрифту, яке оголошувало нелатинський діапазон символів, погіршувало skip-ink для всієї родини. Виправленням стала перейменована родина у стеку шрифтів. Розповідь про підкреслення в Safari і підмножину шрифту, яка його зіпсувала завершується думкою, з якої ця стаття почала б: вимір skip-ink правдивий лише для того рушія, в якому його зроблено. Таблиця, діагноз і перше виправлення стосувалися тільки Chromium, а знімок екрана, через який справу відкрили знову, походив від людини, що просто дивилася на сторінку.

Друк — ще одне місце, де рушії розходяться. Стильовий файл для друку на 637 рядків описує чотири особливості рушіїв як пастки, і за кожною стоїть справжній дефект. Одиниці viewport в одному рушії рахуються від сторінкового блока, а в іншому — від вікна, тож сторінка, надрукована з широкого вікна, втрачала третину кожного рядка. Елементи з фіксованим положенням друкуються на кожному аркуші. Підписи, розфарбовані для типової темної теми, виходили білим по білому в читачів, чий браузер

віддавав перевагу темній. А flex- чи grid-контейнер, який сторінкові носії не вміють розірвати, давав порожній перший аркуш. Перевірка, що це стереже, проходить шість комбінацій паперу й орієнтації за обох колірних схем і рахує фактичні сторінки, а сама сторінка відкрито записує, що один рушій ігнорує контроль висячих рядків.

Третій випадок знайшли читачі, а не тест. Коли сайт вийшов як Tor onion-дзеркало з читабельною адресою, його перші читачі сиділи на іншому рушії, де немає анімації за прокручуванням, на яку розраховував футер, тож колофон лягав поверх головної сторінки в кожного з них. Дефект був і на звичайному сайті. Просто знадобилася аудиторія на іншому рушії, щоб його показати.

Де виникають проблеми

- **Сторінку перевіряють у браузері на столі розробника й вважають готовою.** Кожна звичка цього рушія стає прихованим припущенням. Ліки — коротка матриця з одним браузером на кожен рушій плюс телефон, а виправлення вважають завершеним лише після вимірювання в кожному з них.
- **Код читає назву браузера й вгадує, що той уміє.** Вгадування хибить для кожного браузера, що позичає чужу мітку, і застаріває з кожним випуском. Краще питати про саму можливість. Правило поза блоком дістанеться всім іншим рушіям, тож воно має описувати безпечний стан:

```
.footer {
    position: static;
}

@supports (animation-timeline: scroll()) {
    .footer {
        animation: settle linear both;
        animation-timeline: scroll();
    }
}
```

- **Елементи без стилів успадковують типові стилі браузера.** Заголовки, відступи й особливо елементи форм приходять із вбудованою таблицею стилів кожного рушія, а вони різні. Дизайн, що здавався завершеним, міг спиратися на вибір одного рушія, зроблений без чийогось рішення.

- **Правило, яке діє лише на сторінці, вважають чинним.** Кнопка, яку скрипт зробив сірою, поле, що приймає тільки цифри, посилання, сховане від людей без підписки, — кожне з них тримається лише доти, доки читач лишає сторінку такою, якою її надіслали. Сервер повторює кожну перевірку, що має значення.
- **Посилання відкриваються всередині чужого застосунку.** Посилання з чату чи соцмережі часто потрапляє у вбудований браузер або webview, нерідко з власним сховищем, тож людина, яка увійшла у свій звичайний браузер, приходить туди як незнайомець. Сценарії, що передають керування іншому застосунку або спираються на наявну сесію, варто перевіряти й з цих точок входу.

Що почитати далі

- NIST CSRC Glossary: web browser — два короткі визначення з публікацій NIST, одне з них — просто «client software used to view Web content».
- HTML Standard: Parsing HTML documents — правила, за якими кожен браузер перетворює розмітку, правильну чи ні, на те саме дерево.
- HTML Standard: Event loops — як введення, скрипти, відповіді мережі й малювання сторінки виконуються по черзі.
- HTML Standard: Loading web pages — походження, сайти й правила, що не дають одній сторінці читати іншу.
- WebKit Bugzilla: bug 255159 — власний запис рушія про ваду skip-ink, описану вище.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке браузер?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/browser/