Engineer

Стрічка

оновлення сторінки у форматі для підписки

Що це таке

Стрічка — це файл за сталою адресою, у якому перелічено найновіші матеріали сайту у вигляді, що його може прочитати програма: кожен запис має заголовок, дату, ідентифікатор, який ніколи не змінюється, і зазвичай сам текст. Програма завантажує цей файл за розкладом і показує людині лише те, що з'явилося відтоді. Її також називають вебстрічкою або фідом, а багато хто каже «RSS-стрічка» про будь-яку з них, хоч би який формат лежав усередині.

Різниця — у напрямку руху. Розсилка приходить сама: відправник тримає список адрес і надсилає лист на кожну. Стрічку забирають: видавець кладе один файл на сервер, а програма для читання стрічок у кожного підписника завантажує його, порівнює з копією, яку бачила минулого разу, і залишає записи, яких раніше не зустрічала. Видавець ніколи не дізнається, хто підписався, підписник ніколи не віддає адреси електронної пошти, а щоб відписатися, досить видалити один рядок у своїй програмі.

Майже всі стрічки в мережі написано в одному з трьох форматів. RSS 2.0 — це XML, дати записів у ньому пишуть у стилі RFC 822 (`Sat, 07 Sep 2002 00:00:01 GMT`), а власна специфікація оголошує формат замороженим. Atom, який IETF опублікувала як RFC 4287 у грудні 2005 року, теж є XML і суворіший: кожна стрічка й кожен запис мають сталий `id` і позначку часу `updated` у форматі RFC 3339, а автора треба вказати хоча б в одному місці. JSON Feed, з серпня 2020 року у версії 1.1, каже те саме мовою JSON для розробників, які воліють не розбирати XML.

Сторінка повідомляє програмам, де лежать її стрічки, елементами `<link rel="alternate">` у розділі `head` , кожен зі своїм медіатипом: `application/rss+xml` , `application/atom+xml` або `application/feed+json` . Так

програма знаходить стрічку, маючи лише звичайну адресу сайту.

Застосуйте поняття до рішення

Застосуйте «Що таке стрічка?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Швидкий сайт, який легко знайти в пошуку і який ваші працівники редагують самі, — без рахунку за кожну зміну в тексті.

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. Розробила 100 вебзастосунків на HTML/HTML5 — 100 вебзастосунків побудовано

Розробила 100 вебзастосунків на HTML/HTML5 Зв'язатися з нами →

Чому це важливо

Перше рішення — чи публікувати стрічку взагалі. Це ще один вихідний формат, який треба описати шаблоном, перевірити валідатором і тримати незмінним роками, для аудиторії, меншої за аудиторію соцмережі. Натомість це єдиний канал підписки, якому не потрібні ні платформа, ні обліковий запис, ні список розсилки, і між публікацією та людьми, які просили її показувати, не стоїть жоден алгоритм ранжування.

Друге рішення — які формати. Одні програми читають лише RSS, інші лише Atom, і наперед неможливо знати, якого хоче програма конкретного підписника. Публікація обох закриває цю прогалину ціною другого шаблону, а JSON Feed додає третій для програм, написаних людьми, яким зручніше працювати з JSON. Кожен додатковий формат — ще один файл, який треба узгоджувати з іншими: ті самі записи, ті самі ідентифікатори, ті самі дати.

Третє — скільки тексту класти в кожен запис. Стрічка з повним текстом дає змогу прочитати весь матеріал у програмі для читання, і саме для цього такі програми здебільшого й існують, але видавець утрачає перегляди сторінок. Стрічка лише з анотаціями відправляє читача на сайт по кожен запис. Обидва варіанти прийнятні; вибір — між тим, як далеко подорожує текст, і тим, як часто відвідують саму сторінку.

Хоч якими будуть відповіді, адреса — це зобов’язання. Програми опитують ту URL-адресу, яку отримали, за розкладом, що його кожна обирає сама (RSS дозволяє стрічці запропонувати інтервал у хвилинах через `ttl`, і програма може його проігнорувати), і багато хто опитує її роками. Щойно адресу стрічки роздано, вона залишається.

Як це виглядає на практиці

Цей сайт публікує всі три формати зі своєї української головної сторінки: `/uk/index.xml` (RSS), `/uk/atom.xml` (Atom) і `/uk/feed.json` (JSON Feed). Сторінка подяк перелічує їх і сервіси, що їх перевіряють: усі три містять повний текст, а XML-стрічок дві, а не одна, бо деякі програми для читання стрічок підтримують лише один із двох форматів, і неможливо передбачити, який саме. Валідатор стрічок W3C перевіряє стрічку Atom і стрічку RSS, а валідатор JSON Feed — третю, якої сервіс W3C не охоплює, бо він старший за цей формат. Сервіс W3C попереджає, що більшість записів мають однакову дату зміни. Сторінка показує це попередження, а не вигадує дату: дати беруться з історії git, і більшість сторінок справді написано в тих самих кількох комітах.

Сторінка контактів пропонує стрічки як спосіб стежити за компанією без облікового запису, поруч із профілями компанії, і повторює, що всі три містять повний текст усього опублікованого.

Жодну з трьох не пишуть вручну й не створюють на запит. У проєкті, де побудовано цей сайт, сайт генерується цілком під час збирання, без бекенду часу виконання, кожну сторінку публікують у кількох поданнях з одного дерева вмісту, а головна сторінка додає до цього три стрічки та маніфест встановлюваного застосунку.

Де виникають проблеми

Усі старі записи знову з'являються як нові. Після редизайну, переїзду на інший домен або зміни шаблону підписники бачать десятки записів, які прочитали кілька місяців тому. Програми впізнають запис за ідентифікатором — `id` в Atom, `guid` в RSS або `id` в JSON Feed, — і коли цей рядок змінюється, запис для них новий. RFC 4287 прямо вимагає, щоб `id` запису пережив переміщення, перехід на іншу систему й повторну публікацію. Постійне посилання годиться на ідентифікатор лише доти, доки сайт обіцяє зберігати старий рядок після переїзду. У RSS варто ще й прямо сказати, чим є `guid`: специфікація за замовчуванням вважає `isPermaLink` істинним, а RSS-шаблон цього сайту все одно пише `isPermaLink="true"`, бо програма, яка вгадає навпаки, імпортує кожен запис повторно як дублікат.

Стрічка стверджує, що змінилася після кожного збирання. Дата на рівні стрічки зсувається щоразу, коли сайт збирають, байдуже, чи щось опубліковано, тож програми, що стежать за цією датою, спрацьовують без приводу. Причина — шаблон, який під час збирання ставить поточний час. RSS визначає `lastBuildDate` як час останньої зміни вмісту каналу, а `updated` на рівні стрічки в Atom означає те саме, тож дата має братися з найновішого запису. Atom-шаблон цього сайту бере її з найновішого запису саме з цієї причини.

XML-парсер перестає читати стрічку після типографської зміни. Програми повідомляють, що стрічка зламана, а валідатор вказує на невизначену сутність у першій статті з типографськими лапками. XML має п'ять наперед визначених іменованих сутностей — `lt` , `gt` , `amp` , `apos` і `quot` , — тоді як HTML визначає їх сотні. Рендерер Markdown, що записує лапки й тире як іменовані сутності HTML, створює розмітку, яка на сторінці працює, а всередині елемента XML фатальна. Екрануйте HTML перед вставкою, як і очікує вміст Atom із `type="html"` , або загорніть його в секцію CDATA, яку RSS Best Practices Profile дозволяє для `content:encoded` . Atom-шаблон цього сайту екранує свій вміст, а коментар поруч фіксує, що в сирому вигляді стрічка не розбиралася зовсім.

Підписники тихо зникають після переїзду. Стрічка не має кнопки «відписатися» і списку, що скорочується, тож єдиний знак — тиша. Адреса стрічки змінилася або файл прибрали, і кожна програма, що опитує стару URL-адресу, тепер отримує помилку. Тримайте стару адресу живою — або з самою стрічкою, або з постійним перенаправленням. Цей сайт тримає свої RSS-стрічки на `index.xml` , стандартному шляху генератора, яким його зібрано, бо саме цю адресу опитують наявні програми; а мовою, якої сайт не оголошує, він однаково пише правильно сформовані стрічки з нулем записів, щоб програма отримала дійсний документ, а не 404.

Дві стрічки зливаються в одну в діалозі підписки. Розширення браузера або програма для читання пропонує одну стрічку там, де сторінка оголошує дві. Два альтернативні посилання з однаковим типом і однаковою назвою для такого інструмента виглядають як одна стрічка, і він показує ту, яку побачив останньою. Дайте кожній власну назву; сторінки цього сайту називають свої стрічки назвою сайту, після якої стоїть «(RSS)», «(Atom)» або «(JSON Feed)».

Що почитати далі

- RFC 4287: The Atom Syndication Format — стандарт IETF з обов’язковими елементами, сталими ідентифікаторами й датами за RFC 3339.
- RSS 2.0 Specification — текст замороженого формату від RSS Advisory Board, елемент за елементом.
- JSON Feed Version 1.1 — та сама ідея мовою JSON і що змінилося порівняно з версією 1.0.
- HTML Standard: link type alternate — як сторінка вказує програмам на свої стрічки.
- W3C Feed Validation Service — перевіряє стрічку RSS або Atom за її адресою й пояснює кожне попередження.

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 стрічка?» 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/feed/