Engineer

URL

адреса, що каже, де саме лежить річ

Що це таке

URL (Uniform Resource Locator), який у побуті називають вебадресою або просто посиланням, — це рядок тексту, що водночас називає ресурс і пояснює програмі, як його отримати: яким протоколом говорити, до якої машини звертатися і що саме в неї просити. Вставте його в браузер, термінал чи конфігураційний файл, і кожен із них зможе діяти без додаткових підказок.

Будову URL задає загальний синтаксис, спільний для всіх схем:

```
https://shop.example.com:8443/orders/42?view=full#items

схема      https
хост       shop.example.com
порт       8443
шлях       /orders/42
запит      view=full
фрагмент   items
```

Схема каже, як дістатися до ресурсу, тут це HTTP поверх TLS. Хост і необов'язковий порт кажуть, куди звертатися, шлях — який саме ресурс на цьому хості, а рядок запиту несе параметри, зазвичай парами `назва=значення`, розділеними знаком `&`. Фрагмент після `#` вказує всередину результату: на заголовок чи певне місце на сторінці. Він належить клієнтові: браузер не включає його до цільової частини HTTP-запиту (request-target). Це не робить його таємним для сервера, адже клієнтський код може його прочитати й явно надіслати в іншому запиті.

Символи `/` , `?` , `#` , `&` і `=` мають у цій структурі свої ролі. Дані, що конфліктували б із роздільником у компоненті, який формується, треба закодувати відсотками, тобто записати як `%` і дві шістнадцяткові цифри. Які саме символи потребують цього, залежить від компонента та його синтаксису. Пробіл стає `%20` , амперсанд усередині значення параметра запиту — `%26` , а сам знак відсотка — `%25` . Літери поза ASCII передаються так само, у вигляді своїх байтів UTF-8, і саме так українські слова в шляху доходять до сервера.

Чи URL і URI — одне й те саме? Майже. URI (Uniform Resource Identifier) — це будь-який рядок у цьому синтаксисі, що ідентифікує ресурс. URL — різновид URI, який ще й каже, де ресурс узяти, тоді як `urn:isbn:9780000000002` називає книжку й не дає способу її отримати. RFC 3986 просить нові специфікації вживати URI. Стандарт URL від WHATWG, якого дотримуються браузери, пішов протилежним шляхом: обрав URL єдиним терміном, описує те, що насправді роблять реалізації, і має на меті замінити RFC. На звичайних адресах ці два документи збігаються, а в граничних випадках розходяться, і про це варто пам’ятати, коли браузер і серверна бібліотека розбирають той самий дивний рядок.

Застосуйте поняття до рішення

Застосуйте «Що таке URL?» 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

Опублікована URL-адреса — це обіцянка. Вона потрапляє в закладки, пошукові індекси, листи, друковані буклети, історію чатів і на чужі сторінки, а жодна з цих копій сама не оновлюється. Якщо адреса згодом зміниться, зламаються всі вони, хіба що стара адреса й надалі відповідатиме перенаправленням на нову.

Тому ключове рішення ухвалюють на самому початку: з чого будувати адресу. Усе, що найпевніше зміниться, — поганий фундамент: заголовок сторінки, дерево категорій, назва продукту чи розширення файлу, яке видає

поточний фреймворк. Короткий ідентифікатор або slug, збережений як дані й змінюваний лише свідомо, живе довго. Так само потрібен план на випадок, коли адреса все ж мусить переїхати: стара стає постійним перенаправленням, а не сторінкою 404.

Друге рішення стосується того, чого в адресі бути не повинно. RFC 9110 каже про це прямо: адреси створено, щоб ними ділитися, а не щоб їх захищати. Їх пишуть у журнали серверів і проксі, зберігають в історії браузера, копіюють в аналітику і, залежно від політики, передають далі в полі `Referer`. Ідентифікаторам сесій, токенам скидання пароля, ключам API та персональним даним місце в заголовках або в тілі запиту.

Довжина — менше питання. Специфікація HTTP рекомендує, щоб програми приймали адреси завдовжки щонайменше 8 000 октетів. Це щедро, але рядок запиту, що несе цілу форму чи серіалізований фільтр, зазвичай раніше впирається в коротше обмеження десь у ланцюжку.

Як це виглядає на практиці

У роботі над цим сайтом адреси розглядають як дані, що живуть довше за текст на сторінці. Коли 495 сторінок досягнень трьома мовами згенерували з експорту SQLite, відкритого лише для читання на задокументованому етапі реалізації в серпні 2026 року, найважливіше рішення стосувалося саме адреси. До того сайт виводив URL сторінки з перших слів її англійського твердження, тож виправлене речення непомітно переносило сторінку й ламало кожне посилання на неї. Після цього адресу записували як дані, по рядку на кожну адресу кожного досягнення: перший рядок був поточною адресою, а кожен наступний — виведеною з ужитку адресою, яку сайт видавав як перенаправлення. Зафіксований результат полягав у тому, що всі адреси, опубліковані до того етапу, і далі відповідали.

Нотатка про мобільні гаманці показує, що відбувається, коли одна URL-адреса їде всередині іншої. Phantom, Solflare і Backpack відкривають dApp через універсальні посилання, які несуть цільову адресу у власному шляху, а джерело запиту — в параметрі `ref`, і обидва значення проходять через `encodeURIComponent`. Незакодований `?` або `&` у цільовій адресі ламає розбір, і гаманець відкривається на головному екрані замість вкладки браузера. Із вигаданою власною схемою на кшталт `backpack://` ще гірше: застосунок не реєструє для неї маршруту, і користувач опиняється на сторінці встановлення.

Адреси працюють і як ідентичність. Коли 963 блоки структурованих даних замінили одним зв'язаним графом, ідентифікатори організації, сайту й людини утворили від походження сайту, а сторінки переліків публікують свої елементи лише як URL. На задокументованому етапі реалізації в липні 2026 року це рішення виміряли: назвати елементи на індексі портфоліо означало б 107 повних тверджень і стиснуту сторінку, більшу на 52 відсотки. Перевірка, яка стереже граф, стверджує, що кожну URL-адресу, названу в ньому, справді зібрано.

Де виникають проблеми

Помилки в URL рідко гучні. Адреса виглядає правильною, просто веде деінде або розкриває більше, ніж хотілося.

Кодування відбувається нуль разів або двічі. Адреса повернення, вставлена в запит як є, наприклад `?next=/cart?item=4&size=m`, розпадається на незакодованому `&`, і сервер бачить `size=m` як окремий параметр. Якщо закодувати вже закодований рядок, `%20` стає `%2520`, і сторінка отримує буквальне `%20`. Кодуйте кожну частину рівно один раз, у момент, коли вона потрапляє в адресу, і функцією, призначеною саме для частин. У JavaScript це `encodeURIComponent`, а `encodeURI` лишає `?`, `&` і `#` як є, бо розрахований на цілу адресу.

Секрет їде в рядку запиту. Токен скидання пароля чи ключ API в URL потрапляє в журнали доступу, історію браузера й аналітику, а за поблажливої політики реферера — ще й у `Referer` для кожної сторонньої служби, з якої сторінка щось завантажує. Облікові дані кладіть у заголовок. Якщо токен мусить їхати в посиланні, як у листі для скидання пароля, зробіть його одноразовим і короткоживучим та задайте `Referrer-Policy`, яка не випускає його назовні.

Одна сторінка відповідає за п'ятьма адресами. `http` і `https`, з `www` і без, із кінцевою скісною рискою і без неї, `/About` і `/about`. Схема та хост за визначенням нечутливі до регістру, а шлях чутливий, якщо сервер не вирішить інакше. Пошукові системи ділять вагу сторінки між копіями, кеші зберігають кожну окремо, аналітика рахує їх нарізно. Оберіть одну форму, постійно перенаправляйте на неї всі інші й позначте її як канонічну.

Відносне посилання розв’язується від хибної бази. Посилання на кшталт `img/logo.png` замінює все після останньої скісної риски поточного шляху. На `/docs/guide` воно веде до `/docs/img/logo.png`, а на `/docs/guide/` — до `/docs/guide/img/logo.png`. Одна кінцева риска зсуває всі відносні посилання на сторінці. Посилання, що починаються з `/`, цю проблему обходять.

Логіка спирається на фрагмент. Перенаправлення на сервері, журнали доступу й аналітика, що читають цільову частину HTTP-запиту (request-target), не отримують ні `#`, ні фрагмента після нього. Клієнт може надіслати його окремо, коли цього потребує застосунок. Те, що має дійти до сервера в складі початкового запиту, повинно стояти в шляху або в рядку запиту.

Що почитати далі

- RFC 3986: Uniform Resource Identifier (URI): Generic Syntax — граматика, кодування відсотками, розв’язання відносних посилань і нормалізація.
- WHATWG URL Standard — алгоритм розбору, який реалізують браузери, крок за кроком.
- RFC 3987: Internationalized Resource Identifiers — як адреси, записані будь-якою писемністю, перетворюються на закодовані байти.
- RFC 8820: URI Design and Ownership — чому вигляд шляхів на хості визначає його власник, і тільки він.
- RFC 9110, розділ 17.9 — застереження щодо чутливих даних в адресах.

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 URL?» 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/url/