Engineer

URL

die Adresse, die angibt, wo etwas zu finden ist

Was es ist

Eine URL, kurz für Uniform Resource Locator und im Alltag Webadresse oder einfach Link genannt, ist eine Textzeile, die eine Ressource bezeichnet und der Software zugleich sagt, wie sie abzurufen ist: welches Protokoll sie sprechen, welchen Rechner sie ansprechen und was sie dort anfordern soll. In einen Browser, ein Terminal oder eine Konfigurationsdatei eingefügt, lässt sie sich dort jeweils ohne weitere Hilfe verwenden.

Ihr Aufbau folgt der allgemeinen Syntax, die alle Schemata gemeinsam haben:

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

scheme     https
host       shop.example.com
port       8443
path       /orders/42
query      view=full
fragment   items
```

Das Schema sagt, wie die Ressource erreicht wird, hier über HTTP mit TLS. Host und optionaler Port sagen, wo; der Pfad bezeichnet die Ressource auf diesem Host, und der Abfrageteil trägt Parameter, meist als Paare der Form `name=value`, verbunden durch `&`. Das Fragment hinter `#` verweist auf eine Stelle im Ergebnis, etwa eine Überschrift oder eine Position. Es gehört zum Client: Der Browser lässt es aus dem HTTP-Anfrageziel weg. Dadurch bleibt es dem Server nicht geheim, denn clientseitiger Code kann es lesen und ausdrücklich in einer anderen Anfrage senden.

Zeichen wie `/`, `?`, `#`, `&` und `=` haben in diesem Aufbau bestimmte Aufgaben. Daten, die mit einem Trennzeichen innerhalb der gerade aufgebauten Komponente kollidieren würden, müssen deshalb prozentkodiert werden: als `%` mit zwei folgenden Hexadezimalziffern. Welche Zeichen das betrifft, hängt von der Komponente und ihrer Syntax ab. Ein Leerzeichen wird zu `%20`, ein Und-Zeichen innerhalb eines Abfrageparameterwerts zu `%26` und das Prozentzeichen selbst zu `%25`. Buchstaben außerhalb von ASCII werden ebenso als ihre UTF-8-Bytes übertragen; so kommen dänische oder ukrainische Wörter in einem Pfad an.

Ist eine URL dasselbe wie eine URI? Fast. Eine URI, ein Uniform Resource Identifier, ist jede Zeichenfolge in dieser Syntax, die etwas identifiziert. Eine URL ist die Art, die zusätzlich angibt, wie es zu finden ist, während `urn:isbn:9780000000002` ein Buch benennt, ohne einen Weg zum Abruf anzugeben. RFC 3986 fordert neue Spezifikationen auf, URI zu sagen. Der URL Standard von WHATWG, dem Browser folgen, ging den umgekehrten Weg: Er legte URL als einzigen Begriff fest und beschreibt das Verhalten tatsächlicher Implementierungen mit dem Ziel, den RFC abzulösen. Bei gewöhnlichen Adressen stimmen beide überein, in Grenzfällen unterscheiden sie sich. Das sollte im Blick bleiben, wenn ein Browser und eine Serverbibliothek dieselbe ungewöhnliche Zeichenfolge auswerten.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine URL?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Eine schnelle, auffindbare Website, die Ihre Mitarbeitenden selbst bearbeiten können, ohne für jede Textänderung eine Rechnung zu bekommen.

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. 100 Webanwendungen mit HTML/HTML5 entwickelt — 100 entwickelte Webanwendungen

100 Webanwendungen mit HTML/HTML5 entwickelt Kontakt aufnehmen →

Warum es wichtig ist

Eine veröffentlichte URL ist ein Versprechen. Sie landet in Lesezeichen, Suchindizes, E-Mails, gedruckten Broschüren, Chatprotokollen und den Seiten anderer Leute. Keine dieser Kopien aktualisiert sich selbst. Eine spätere Änderung der Adresse macht sie alle unbrauchbar, sofern die alte Adresse nicht weiter mit einer Weiterleitung zur neuen antwortet.

Deshalb lohnt sich eine frühe Entscheidung darüber, woraus die Adresse gebildet wird. Alles, was sich wahrscheinlich ändern wird, ist eine schlechte Grundlage, ob Seitentitel, Kategoriebaum, Produktname oder eine Dateiendung, die das aktuelle Framework verrät. Eine kurze Kennung oder ein als Daten gespeicherter und bewusst bearbeiteter Slug hält länger. Ebenso wichtig ist ein Plan für den Fall, dass eine Adresse trotzdem umziehen muss: Die alte wird zu einer dauerhaften Weiterleitung statt zu einer 404.

Die zweite Entscheidung betrifft das, was draußen bleibt. RFC 9110 sagt es deutlich: Adressen sind zum Teilen gedacht, nicht zur Geheimhaltung. Sie werden in Server- und Proxyprotokolle geschrieben, im Browserverlauf gespeichert, in Analysedaten kopiert und je nach Richtlinie im Header `Referer` an das weitergegeben, was die Seite als Nächstes lädt. Sitzungskennungen, Tokens zum Zurücksetzen, API-Schlüssel und personenbezogene Daten gehören in Header oder Anfragekörper.

Die Länge ist eine weniger wichtige Frage. Die HTTP-Spezifikation empfiehlt Software, Adressen von mindestens 8.000 Oktetten anzunehmen. Das ist großzügig, doch ein Abfragestring mit einem ganzen Formular oder einem serialisierten Filter stößt entlang der Kette meist schon vorher auf eine kürzere Grenze.

Wie es in der Praxis aussieht

Bei der Arbeit an dieser Website werden Adressen als Daten behandelt, die den Text der Seite überdauern. Als auf dem dokumentierten Implementierungsstand im August 2026 495 Seiten mit Arbeitsergebnissen in drei Sprachen aus einem schreibgeschützten SQLite-Export erzeugt wurden, betraf die folgenreichste Entscheidung die Adresse. Die Website hatte die URL einer Seite aus den ersten Wörtern ihrer englischen Aussage abgeleitet. Eine Satzkorrektur verschob die Seite dadurch unbemerkt und brach jeden Link zu ihr. Danach wurde die Adresse als Daten gepflegt, eine Zeile je Adresse und Arbeitsergebnis: Die erste Zeile war die aktuelle Adresse, jede spätere eine ausgemusterte, die die Website als Weiterleitung ausgab. Das dokumentierte Ergebnis war, dass alle bis zu diesem Stand veröffentlichten Adressen weiter antworteten.

Eine Notiz zu mobilen Wallets zeigt, was passiert, wenn eine URL innerhalb einer anderen transportiert wird. Phantom, Solflare und Backpack öffnen eine dApp über universelle Links, die die Zieladresse im eigenen Pfad und den anfragenden Ursprung in einem `ref` -Parameter tragen, jeweils durch `encodeURIComponent`

kodiert. Ein nicht kodiertes `?` oder `&` in der Zieladresse zerstört die Auswertung, und die Wallet öffnet ihre Startansicht statt des Browser-Tabs. Ein auf Verdacht gewähltes benutzerdefiniertes Schema wie `backpack://` ist noch schlechter: Die Anwendung registriert keine dazu passende Route, und der Nutzer landet auf der Installationsseite.

Adressen dienen auch als Identität. Als 963 Blöcke strukturierter Daten durch einen verknüpften Graphen ersetzt wurden, wurden die Kennungen für Organisation, Website und Person aus dem Ursprung der Website gebildet. Listenseiten veröffentlichen ihre Einträge nur als URLs. Auf dem dokumentierten Implementierungsstand im Juli 2026 wurde diese Entscheidung vermessen: Die Einträge auf der Portfolioübersicht zu benennen hätte 107 vollständige Aussagen und eine um 52 Prozent größere komprimierte Seite bedeutet. Die Prüfung des Graphen stellt sicher, dass jede darin genannte URL beim Website-Build tatsächlich erzeugt wurde.

Wo es schiefgeht

URL-Fehler bleiben oft unauffällig. Die Adresse sieht weiterhin richtig aus; sie zeigt lediglich woandershin oder verrät mehr als vorgesehen.

Die Kodierung erfolgt gar nicht oder zweimal. Eine unverändert in eine Abfrage eingefügte Rücksprungadresse wie `?next=/cart?item=4&size=m` wird am nicht kodierten `&` aufgeteilt. Der Server sieht `size=m` dann als eigenen Parameter. Wird eine bereits kodierte Zeichenfolge erneut kodiert, wird aus `%20` ein `%2520`, und die Seite erhält ein wörtliches `%20`. Jede Komponente sollte genau einmal kodiert werden, wenn sie in die Adresse eingesetzt wird, und zwar mit der für Komponenten vorgesehenen Funktion. In JavaScript ist das `encodeURIComponent`; `encodeURI` lässt `?`, `&` und `#` stehen, weil es eine vollständige Adresse erwartet.

Ein Geheimnis reist im Abfragestring mit. Ein Token zum Zurücksetzen eines Passworts oder ein API-Schlüssel in der URL wird in Zugriffsprotokolle, Browserverlauf und Analysedaten kopiert. Bei einer großzügigen Referrer-Richtlinie wird er außerdem im `Referer` an jeden Drittanbieter gesendet, von dem die Seite etwas lädt. Zugangsdaten gehören in einen Header. Muss ein Token in einem Link transportiert werden, etwa in einer E-Mail zum Zurücksetzen, sollte es nur einmal und nur kurz gültig sein. Eine passende `Referrer-Policy` hält es auf der eigenen Seite.

Eine Seite antwortet unter fünf Adressen. `http` und `https`, mit und ohne `www`, mit und ohne abschließenden Schrägstrich, `/About` und `/about`. Bei Schema und Host spielt Groß- und Kleinschreibung definitionsgemäß keine Rolle; beim Pfad dagegen schon, sofern der Server nichts anderes festlegt. Suchmaschinen verteilen das Gewicht der Seite auf ihre Kopien, Caches speichern jede einzeln und Analysesysteme zählen sie getrennt. Eine Form sollte festgelegt, jede andere dauerhaft dorthin weitergeleitet und die gewählte als kanonisch angegeben werden.

Ein relativer Link wird gegen die falsche Basis aufgelöst. Eine Referenz wie `img/logo.png` ersetzt alles hinter dem letzten Schrägstrich des aktuellen Pfads. Bei `/docs/guide` wird daraus `/docs/img/logo.png`, bei `/docs/guide/` dagegen `/docs/guide/img/logo.png`. Ein einziger abschließender Schrägstrich verschiebt jeden relativen Link auf der Seite. Von der Wurzel ausgehende Links, die mit `/` beginnen, umgehen das Problem.

Die Logik hängt vom Fragment ab. Serverseitige Weiterleitungen, Zugriffsprotokolle und Analysesysteme, die das HTTP-Anfrageziel lesen, erhalten weder `#` noch das Fragment. Ein Client kann es gesondert senden, wenn die Anwendung es braucht. Was den Server bereits als Teil der ersten Anfrage erreichen muss, gehört in den Pfad oder den Abfrageteil.

Weiterführende Lektüre

- RFC 3986: Uniform Resource Identifier (URI): Generic Syntax — die Grammatik, Prozentkodierung, Auflösung relativer Referenzen und Normalisierung.
- WHATWG URL Standard — der von Browsern implementierte Parsing-Algorithmus, Schritt für Schritt.
- RFC 3987: Internationalized Resource Identifiers — wie Adressen in beliebigen Schriftsystemen auf prozentkodierte Bytes abgebildet werden.
- RFC 8820: URI Design and Ownership — warum allein der Inhaber eines Hosts bestimmt, wie dessen Pfade aussehen.
- RFC 9110, Abschnitt 17.9 — der Sicherheitshinweis zu vertraulichen Informationen in Adressen.

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 eine URL?“ 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/url/