Engineer

URL

〒 l'adresse qui indique où trouver quelque chose

Ce que c'est

Une URL, abréviation d'Uniform Resource Locator et appelée couramment adresse web ou simplement lien, est une ligne de texte qui désigne une ressource tout en indiquant au logiciel comment la récupérer : quel protocole utiliser, à quelle machine s'adresser et quoi lui demander. Collée dans un navigateur, un terminal ou un fichier de configuration, elle permet à chacun d'agir sans autre aide.

Sa structure est fixée par la syntaxe générique commune à tous les schémas :

```
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
```

Le schéma indique comment atteindre la ressource, ici par HTTP sur TLS. L'hôte et le port facultatif indiquent où, le chemin désigne la ressource sur cet hôte et la partie requête transporte des paramètres, généralement des paires `name=value` reliées par `&`. Le fragment après `#` désigne un endroit dans le résultat, un titre ou une position. Il relève du client : le navigateur l'omet de la cible de la requête HTTP. Cela ne le rend pas secret pour le serveur, car du code côté client peut le lire et l'envoyer explicitement dans une autre requête.

Puisque des caractères comme `/`, `?`, `#`, `&` et `=` jouent un rôle dans cette structure, les données qui entreraient en conflit avec un délimiteur dans le composant en cours de construction doivent être codées en pourcentage : un `%` suivi de deux chiffres hexadécimaux. Les caractères concernés dépendent du composant et de sa syntaxe. Un espace devient `%20`, une esperluette dans la valeur d'un paramètre de requête devient `%26` et le signe pour cent lui-même devient `%25`. Les lettres hors ASCII voyagent de la même façon, sous la forme de leurs octets UTF-8, ce qui permet de transmettre des mots danois ou ukrainiens dans un chemin.

Une URL est-elle la même chose qu'une URI ? Presque. Une URI, ou Uniform Resource Identifier, est toute chaîne respectant cette syntaxe qui identifie quelque chose. Une URL est le type d'URI qui indique aussi comment le localiser, tandis que `urn:isbn:9780000000002` désigne un livre sans donner de moyen de le récupérer. La RFC 3986 demande aux nouvelles spécifications d'employer URI. L'URL Standard de WHATWG, que suivent les navigateurs, a pris la direction opposée : il a retenu URL comme terme unique et entrepris de décrire le comportement des implémentations réelles, dans l'objectif de remplacer la RFC. Les deux concordent sur les adresses ordinaires et divergent sur les cas limites, ce qu'il faut garder à l'esprit lorsqu'un navigateur et une bibliothèque côté serveur analysent la même chaîne inhabituelle.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'une URL ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Un site rapide, facile à trouver et modifiable par votre équipe, sans facture à chaque changement de texte.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 100 applications web avec HTML/HTML5 — 100 applications web développées

100 applications web avec HTML/HTML5 Contactez-nous →

Pourquoi c'est important

Une URL publiée est une promesse. Elle finit dans des favoris, des index de recherche, des courriels, des brochures imprimées, des historiques de conversation et des pages d'autres personnes, et aucune de ces copies ne se met à jour seule. Changer l'adresse plus tard les casse toutes, sauf si l'ancienne continue de répondre par une redirection vers la nouvelle.

La décision utile se prend donc tôt : sur quoi construire l'adresse ? Tout élément susceptible de changer est une mauvaise base, qu'il s'agisse du titre d'une page, d'une arborescence de catégories, d'un nom de produit ou d'une extension de fichier qui révèle le framework actuel. Un identifiant court, ou un slug stocké comme donnée et modifié délibérément, perdure. Il en va de même d'un plan pour le cas où une adresse doit malgré tout déménager : l'ancienne devient une redirection permanente plutôt qu'une 404.

La deuxième décision porte sur ce qui reste à l'extérieur. La RFC 9110 est explicite : les adresses sont faites pour être partagées, pas sécurisées. Elles sont écrites dans les journaux des serveurs et des proxys, conservées dans l'historique du navigateur, copiées dans les données d'analyse et, selon la politique, transmises dans l'en-tête `Referer` à ce que la page charge ensuite. Les identifiants de session, les jetons de réinitialisation, les clés d'API et les données personnelles ont leur place dans les en-têtes ou les corps de requête.

La longueur est une question secondaire. La spécification HTTP recommande aux logiciels d'accepter des adresses d'au moins 8 000 octets, ce qui est généreux. Mais une chaîne de requête contenant tout un formulaire ou un filtre sérialisé tend à rencontrer une limite plus courte quelque part dans la chaîne avant de les atteindre.

À quoi cela ressemble en pratique

Sur ce site, les adresses sont traitées comme des données qui survivent au texte de la page. Lorsque 495 pages de réalisations en trois langues ont été générées à partir d'un export SQLite en lecture seule au stade d'implémentation documenté d'août 2026, la décision la plus lourde de conséquences concernait l'adresse. Le site avait déduit l'URL d'une page des premiers mots de son énoncé anglais : corriger une phrase déplaçait donc silencieusement la page et cassait tous les liens vers elle. L'adresse a ensuite été rédigée comme donnée, une ligne par adresse et par réalisation : la première ligne était l'adresse actuelle et chaque suivante une ancienne adresse que le site produisait comme redirection. Le résultat consigné était que toutes les adresses publiées jusqu'à ce stade répondaient encore.

Une note sur les portefeuilles mobiles montre ce qui se passe lorsqu'une URL voyage à l'intérieur d'une autre. Phantom, Solflare et Backpack ouvrent une dApp par des liens universels qui portent l'adresse cible dans leur propre chemin et l'origine demandeuse dans un paramètre `ref`, chacune passée à

`encodeURIComponent`. Un `?` ou un `&` non codé dans l'adresse cible corrompt l'analyse, et le portefeuille s'ouvre sur son écran d'accueil au lieu de l'onglet du navigateur. Deviner un schéma personnalisé, comme `backpack://`, donne un résultat encore pire : l'application n'enregistre aucune route correspondante et l'utilisateur arrive sur la page d'installation.

Les adresses servent aussi d'identité. Lorsque 963 blocs de données structurées ont été remplacés par un seul graphe lié, les identifiants de l'organisation, du site et de la personne ont été créés à partir de l'origine du site, et les pages de listes ne publient leurs éléments que par URL. Au stade d'implémentation documenté de juillet 2026, ce choix a été mesuré : nommer les éléments sur l'index du portfolio aurait représenté 107 énoncés complets et une page compressée 52 pour cent plus volumineuse. Le contrôle qui protège le graphe vérifie que chaque URL qu'il nomme a réellement été construite.

Où les problèmes surviennent

Les erreurs d'URL sont souvent discrètes. L'adresse semble toujours correcte ; elle pointe simplement ailleurs ou révèle plus que prévu.

Le codage est omis ou effectué deux fois. Une adresse de retour insérée telle quelle dans une requête, comme `?next=/cart?item=4&size=m`, est coupée à l'esperluette `&` non codée, et le serveur voit `size=m` comme un paramètre distinct. Coder une chaîne déjà codée transforme `%20` en `%2520`, et la page reçoit un `%20` littéral. Chaque composant doit être codé exactement une fois, au moment de son insertion dans l'adresse, avec la fonction prévue pour les composants. En JavaScript, c'est `encodeURIComponent` ; `encodeURI` laisse `?`, `&` et `#` intacts, car il attend une adresse entière.

Un secret voyage dans la chaîne de requête. Un jeton de réinitialisation de mot de passe ou une clé d'API dans l'URL est copié dans les journaux d'accès, l'historique du navigateur et les données d'analyse. Avec une politique de référent permissive, il est envoyé dans `Referer` à chaque tiers auprès duquel la page charge des ressources. Les informations d'authentification ont leur place dans un en-tête. Lorsqu'un jeton doit voyager dans un lien, comme dans un courriel de réinitialisation, il faut le rendre à usage unique et de courte durée, et définir une `Referrer-Policy` qui le garde sur le site.

Une page répond à cinq adresses. `http` et `https`, avec ou sans `www`, avec ou sans barre oblique finale, `/About` et `/about`. Le schéma et l'hôte sont par définition insensibles à la casse ; le chemin y est sensible, sauf décision contraire du serveur. Les moteurs de recherche répartissent le poids de la page entre les copies, les caches stockent chacune séparément et les outils d'analyse les comptent à part. Il faut choisir une forme, rediriger toutes les autres vers elle de façon permanente et la désigner comme canonique.

Un lien relatif est résolu à partir de la mauvaise base. Une référence comme `img/logo.png` remplace tout ce qui suit la dernière barre oblique du chemin actuel. Avec `/docs/guide`, elle devient `/docs/img/logo.png` ; avec `/docs/guide/`, elle devient `/docs/guide/img/logo.png`. Une seule barre oblique finale déplace tous les liens relatifs de la page. Les liens relatifs à la racine, commençant par `/`, évitent le problème.

La logique dépend du fragment. Les redirections côté serveur, les journaux d'accès et les outils d'analyse qui lisent la cible de la requête HTTP ne reçoivent ni `#` ni son fragment. Un client peut l'envoyer séparément lorsque l'application en a besoin. Tout ce qui doit atteindre le serveur dès la requête initiale appartient au chemin ou à la partie requête.

Pour aller plus loin

- RFC 3986 : Uniform Resource Identifier (URI): Generic Syntax — la grammaire, le codage en pourcentage, la résolution relative et la normalisation.
- WHATWG URL Standard — l'algorithme d'analyse implémenté par les navigateurs, étape par étape.
- RFC 3987 : Internationalized Resource Identifiers — la façon dont les adresses écrites dans n'importe quel système d'écriture sont converties en octets codés en pourcentage.
- RFC 8820 : URI Design and Ownership — pourquoi le propriétaire d'un hôte décide seul de la forme de ses chemins.
- RFC 9110, section 17.9 — l'avertissement de sécurité sur les informations sensibles dans les adresses.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'une URL ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/url/