Engineer

URL

資源の所在を示すアドレス

どのようなものか

URLはUniform Resource Locatorの略で、日常的にはWebアドレス、あるいは単にリンクとも呼ばれます。資源を識別すると同時に、取得方法をソフトウェアに伝える一行のテキストです。どのプロトコルを使い、どのマシンに、何を要求するかを示します。ブラウザー、端末、設定ファイルに貼り付ければ、どれも追加の説明なしに扱えます。

構成は、どのスキームにも共通する汎用構文で定まっています。

```
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
```

スキームは到達方法を示し、この例ではTLS上のHTTPです。ホストと省略可能なポートが場所を、パスがそのホスト上のどの資源かを示します。クエリーはパラメーターを運び、通常は `name=value` の組を `&` でつなぎます。`#` の後ろのフラグメントは、見出しや位置など、結果の内部を指します。これはクライアント側の情報なので、ブラウザーはHTTPのリクエストターゲットに含めません。ただし、サーバーに対して秘密になるわけではありません。クライアント側のコードが読み取り、別のリクエストで明示的に送ることはできます。

`/`、`?`、`#`、`&`、`=` などの文字は、この構造で役割を持ちます。そのため、組み立てる部分の区切り文字と衝突するデータは、パーセント符号化が必要です。`%` に続けて十六進数二桁で書きます。どの文字を符号化するかは、その

部分と構文によります。空白は `%20` 、クエリーパラメーターの値の中のアンパサンドは `%26` 、パーセント記号自体は `%25` になります。ASCII外の文字も、UTF-8のバイトとして同じ方法で運びます。パスに含まれるデンマーク語やウクライナ語も、この仕組みで通ります。

URLとURIは同じでしょうか。ほぼ同じですが、区別があります。URI、Uniform Resource Identifierは、この構文で何かを識別する文字列全般です。URLは、その中でも所在を示すものです。一方、 `urn:isbn:9780000000002` は本を識別しますが、取得方法は示しません。RFC 3986は、新しい仕様ではURIという語を使うよう求めています。ブラウザーが従うWHATWG URL標準は逆の方向を選び、URLに用語を一本化し、実装の実際の動作を記述してRFCを置き換えることを目指しました。通常のアドレスでは両者は一致しますが、境界的な例では異なります。ブラウザーとサーバーのライブラリが同じ奇妙な文字列を解析するときには、この点が重要です。

概念を判断に生かす

「URLとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 表示が速く、検索で見つけやすく、文章を変えるたびに料金が発生せず、自社の担当者が編集できるサイトをつくります。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 HTML/HTML5でWebアプリケーション100個を開発 — 100 構築したWebアプリケーション

HTML/HTML5でWebアプリケーション100個を開発 お問い合わせ →

なぜ重要なのか

公開したURLは約束です。ブックマーク、検索インデックス、メール、印刷物、チャットの履歴、他人のページに残り、そのコピーが自動で更新されることはありません。後からアドレスを変えると、元のアドレスが新しい場所へリダイレクトし続けない限り、それらのリンクはすべて切れます。

そこで、早く決めておくべきなのが、アドレスを何から作るかです。変わりやすいものは土台に向きません。ページの題名、カテゴリーの階層、製品名、現在のフレームワークを表す拡張子などが該当します。短い識別子や、データとして保存し、意図して編集するスラッグなら長持ちします。それでも移動が必要になったときの方針も決めます。元のアドレスは404にせず、恒久的なリダイレクトにします。

二つ目の判断は、アドレスに何を入れないかです。RFC 9110は、アドレスは共有のためのもので、保護のためのものではないと明言しています。サーバーやプロキシのログ、ブラウザーの履歴、アクセス解析に記録され、方針によっては、ページが次に読み込む相手へ `Referer` ヘッダーで渡されます。セッション識別子、リセット用トークン、APIキー、個人情報は、ヘッダーやリクエスト本文に入れます。

長さは、それほど大きな問題ではありません。HTTPの仕様は、少なくとも8,000オクテットのアドレスを受け付けるよう実装に勧めています。十分な長さですが、フォーム全体や直列化したフィルターをクエリー文字列に載せると、その長さに達する前に、経路のどこかでもっと短い上限に当たりがちです。

実際の活用例

このサイトの実装では、アドレスを、ページ本文より長く残るデータとして扱います。記録された2026年8月の実装段階で、読み取り専用のSQLiteエクスポートから三言語の実績ページ495件を生成したとき、最も影響が大きかった判断はアドレスでした。それまでは英語の実績文の冒頭からURLを作っていたため、文を直すだけでページが黙って移動し、そこへのリンクがすべて切れました。そこで、実績ごと、アドレスごとに一行を持つデータとして、アドレスを明示的に記述しました。最初の行は現在のアドレスで、その後の行は、サイトがリダイレクトとして出力する旧アドレスです。記録された結果では、その段階までに公開したアドレスはすべて応答し続けました。

モバイルウォレットのノートは、URLを別のURLの中で運ぶ例です。Phantom、Solflare、BackpackはユニバーサルリンクでdAppを開きます。対象のアドレスを自身のパスに、要求元のオリジンを `ref` パラメーターに入れ、それぞれ `encodeURIComponent` を通します。対象に含まれる `?` や `&` を符号化しないと解析が壊れ、ウォレットはブラウザーのタブではなくホーム画面を開きます。`backpack://` のようなカスタムスキームを推測して使うと、さらに悪くなります。アプリには一致する経路の登録がなく、利用者はインストールページへ送られます。

アドレスは識別子にもなります。構造化データ963ブロックを一つのリンクされたグラフへ置き換えたとき、組織、サイト、人物の識別子はサイトのオリジンから作り、一覧ページは各項目をURLだけで示すようにしました。記録された2026年7月の実装段階では、この判断の効果も測定しました。ポートフォリオの一覧に各項目の内容を載せると、完全な実績文107件が必要になり、圧縮後のページが52%大きくなる計算でした。グラフを守る検査は、そこに挙がるURLがすべて実際にビルドされたことを確かめます。

問題が起きるところ

URLの不具合は、目立たない傾向があります。アドレスは正しく見えても、別の場所を指したり、意図以上の情報を明かしたりします。

符号化しないか、二重に符号化する。 戻り先を `?next=/cart?item=4&size=m` のように生のままクエリーへ入れると、符号化していない `&` で分かれ、サーバーは `size=m` を独立したパラメーターとして読みます。すでに符号化した文字列をもう一度符号化すると、 `%20` は `%2520` になり、ページには文字どおりの `%20` が届きます。各部分をアドレスへ入れる時点で、その部分用の関数を使い、ちょうど一度だけ符号化します。JavaScriptなら `encodeURIComponent` です。 `encodeURI` はアドレス全体を受け取る想定なので、 `?` 、 `&` 、 `#` を残します。

秘密情報をクエリー文字列に載せる。 URL内のパスワードリセット用トークンやAPIキーは、アクセスログ、ブラウザーの履歴、アクセス解析に複製されます。リファラーポリシーが緩ければ、ページが読み込むすべての第三者へ `Referer` で送られます。認証情報はヘッダーに入れます。リセット用メールのようにトークンをリンクで運ぶ必要があるときは、一回限りで短時間だけ有効にし、外部へ渡さない `Referrer-Policy` を設定します。

同じページが五つのアドレスで応答する。 `http` と `https` 、 `www` の有無、末尾のスラッシュの有無、 `/About` と `/about` などです。スキームとホストは定義上、大文字と小文字を区別しません。パスは、サーバーが別の判断をしない限り区別します。検索エンジンはページの評価をコピー間に分散し、キャッシュはそれぞれ保存し、アクセス解析も別々に数えます。一つの形を選び、それ以外はすべて恒久的にリダイレクトし、その形を正規URLとして示します。

相対リンクの基準が違う。 `img/logo.png` のような参照は、現在のパスの最後のスラッシュ以降を置き換えます。 `/docs/guide` では `/docs/img/logo.png` に、 `/docs/guide/` では `/docs/guide/img/logo.png` になります。末尾のスラッシュ一つで、ページ上の相対リンクがすべて移動します。 `/` で始まるルート相対リンクなら、この問題を避けられます。

処理がフラグメントに依存する。 HTTPのリクエストターゲットを読むサーバー側のリダイレクト、アクセスログ、アクセス解析には、 `#` もその後のフラグメントも届きません。アプリケーションで必要なら、クライアントが別に送れます。最初のリクエストの一部としてサーバーへ届く必要がある情報は、パスかクエリーに入れます。

参考資料

- RFC 3986：Uniform Resource Identifier (URI): Generic Syntax — 構文、パーセント符号化、相対参照の解決、正規化を定めています。
- WHATWG URL標準 — ブラウザーが実装する解析アルゴリズムを、手順ごとに記述しています。
- RFC 3987：Internationalized Resource Identifiers — どの文字体系で書いたアドレスも、パーセント符号化したバイトへ対応付ける方法です。
- RFC 8820：URI Design and Ownership — パスの形を決めるのが、ほかの誰でもなくホストの所有者である理由を説明します。
- RFC 9110、第17.9節 — アドレスに含む機密情報についてのセキュリティー上の注意です。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「URLとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/url/