Engineer

ウェブアプリマニフェスト

サイトをインストール可能なアプリとして記述するJSONファイル

どのようなものか

ウェブアプリマニフェストは、ウェブサイトをアプリケーションとして記述するJSON文書です。名前、起動時に開くアドレス、アプリに含まれるアドレスの範囲、ウィンドウの表示方法、アプリを表す画像を指定します。ページは、`link`要素の`rel`に`manifest`を指定して、このファイルを指します。インストール機能を持つブラウザーは、それを読み、インストールするアプリの内容を決めます。定義元のW3C仕様も、普段の会話でも、単にマニフェストと呼ぶことが多くあります。

ファイル自体は通常のJSONです。登録されているメディアタイプは`application/manifest+json`、拡張子は`.webmanifest`ですが、仕様はJSONのメディアタイプなら受け付け、多くのサイトは`manifest.json`を配信しています。すべてのメンバーは省略可能です。架空の潮汐表サイト向けの、小さな例を示します。

```
{
    "name": "Harbour Tide Tables",
    "short_name": "Tides",
    "start_url": "/tides/",
    "display": "standalone",
    "background_color": "#ffffff",
    "icons": [
        {
            "src": "/icons/tides-512.png",
            "sizes": "512x512",
            "type": "image/png"
        }
    ]
}
```

見た目とは別に、アプリそのものを決めるメンバーが3つあります。`start_url` は、インストールしたアプリの起動時に開くページの候補です。ブラウザーは無視しても、利用者による変更を認めても構いません。`scope` は、アプリの内側とみなすアドレスの範囲です。省略した場合、仕様は起動URLからファイル名、クエリ、フラグメントを除いたものを使うため、上の例では `/tides/` 配下全体がアプリに含まれます。`id` はアプリの識別情報で、省略すると起動URLを使います。相対指定のアイコンのアドレスと起動URLは、リンク元のページではなく、マニフェスト自身のURLを基準に解決します。一方、相対指定の `id` は、起動URLのオリジンを基準に解決します。

残りは表示に関する指定です。`display` で `fullscreen` 、 `standalone` 、 `minimal-ui` 、 `browser` のいずれかを要求し、対応できないブラウザーは、この順で次の候補に移ります。各アイコンには `purpose` を付けられます。省略時は `any` で、切り抜きに耐えるよう描いた画像は `maskable` 、色を捨ててアルファチャンネルだけを使う画像は `monochrome` です。 `theme_color` は、アプリの周囲のウィンドウに色を付けます。関連するW3Cノートでは、インストール画面やストアが表示できる `description` 、 `categories` 、 `screenshots` も定めています。

概念を判断に生かす

「ウェブアプリマニフェストとは？」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。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

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。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

マニフェストは、サイト側が管理するインストール用メタデータを提供します。仕様は、インストールの判断をブラウザーに委ねています。ブラウザーはインストール手段を提供してもよく、その条件は各ブラウザーが決めます。特定のメンバーを必須にするものもあります。チームは判断材料を提供し、ブラウザーが使い方を決めます。そのため、役に立つ問いは、マニフェストを用意するかどうかより、それによって何を約束するかです。

識別情報は、ファイルの更新後も守るべき約束です。仕様は、ブラウザーがマニフェストを再取得した際、 `id` の一致を、すでにインストールしたアプリのマニフェストを新しいファイルで置き換える合図にすべきだとしています。 `id` を省くと起動URLを使うため、起動ページを移すと、意図せず識別情報も変わりま

す。その場合、ブラウザーは利用者の端末にあるアプリとは別のアプリケーションとして扱うべきことになります。最初のリリースで `id` を固定しておけば、1行でこの危険を避けられます。

スコープは、どこまでをアプリのウィンドウ内で扱うかを決めます。範囲内の移動は、インストールしたウィンドウ内にとどまります。範囲外へのリンクも開けますが、仕様は、アプリの外に出たことをブラウザーが利用者に示すよう想定しています。そのため、サイトより狭いスコープでは、セクション間の普通のリンクまで、アプリから出る操作に見えてしまいます。

見た目の問題が最も起きやすいのはアイコンです。ランチャーはアイコンを円、角丸の四角などに切り抜くため、正方形いっぱいに描いたロゴは端が欠けます。仕様が定めるmaskableアイコンの安全領域は、中央に置いた、短辺の5分の2を半径とする円です。その外側は切り落とされることがあります。monochromeアイコンではアルファチャンネルしか使わないため、不透明な背景の上に描いたアイコンは、塗りつぶした四角になってしまいます。

色の指定が働く範囲は、見た目の印象より狭いものです。`background_color` は、スタイルシートが届くまでの短い時間のためにあります。仕様は、スタイルシートが利用可能になった後に、ブラウザーがこの値を使うことを禁じています。最初の描画の色であって、それ以後の色ではありません。

実際の活用例

このサイトでは、マニフェストを手作業で保管せず、テンプレートから生成します。Hugo設定で `application/manifest+json` に拡張子 `webmanifest` を登録し、各言語のホームページの出力にマニフェストを加えています。各言語が独自のファイルを持ち、各ページのheadがその言語のファイルにリンクします。設定には理由も記されています。共通のマニフェストでは起動URLを1つしか指定できず、デンマーク語の訪問者がサイトをインストールしたところ、英語のホームを開くアプリになってしまったためです。

テンプレートはマニフェストをマップとして組み立て、HugoのJSONエンコーダーに渡します。エスケープとカンマはエンコーダーが扱うため、カンマの抜けたファイルを公開することはありません。アイコンのアドレスはすべてホストのルートから始めます。コメントにも理由があります。`/da/manifest.webmanifest` 内の相対アドレスは `/da/` 配下として解決され、デンマーク語のアイコンがすべて404になるためです。アイコン本体は、用途ごとの3つのフォルダーから読み取

ります。それぞれ36から512ピクセルまで10サイズあり、サイズはファイル名から得るので、生成ツールがサイズを追加しても、マニフェストを編集する必要はありません。

`id` は各言語のホームのアドレスに固定し、 `scope` はオリジン全体にしています。そのため、別の言語へのリンクでも、インストールしたウィンドウ内にとどまります。スクリーンショットの名前は `<lang>-<form_factor>-<WxH>.webp` で、テンプレートは言語、画面形状、サイズの3つをファイル名から読み取ります。英語とデンマーク語には、それぞれ1,280×800の横長画像と780×1,688の縦長画像が1枚ずつあります。スクリーンショットがない言語では、空のリストではなく、`screenshots` メンバー自体を省きます。

クレジットページは、サイトが従う標準の1つとしてW3C仕様を挙げ、誰でも実行できる外部検査の1つとしてPWABuilderを挙げています。これは、Microsoftのコンピューターからマニフェストや、サイトのインストールに必要なほかの要素を読み取るサービスです。

インストールページは、同じファイルを利用者側から説明するものです。各ブラウザーのインストール操作の場所を案内し、Appleのブラウザーが自動的にインストールを勧めることはないと記しています。Firefoxには通常のブラウザーとは別のウェブアプリ用ウィンドウ機能があり、Windowsでは143以降、Microsoft Storeからのインストールでは150以降で、初期状態から有効です。Mozillaの実装文書によれば、Linuxでは初期状態で無効、macOSでは未対応です。このページは、このサイトへの期待も調整しています。インストールされるのは、独自のアイコンとウィンドウを持つアプリです。サイトはサービスワーカーも、管理されたオフライン用コピーも公開していないため、通常の利用には接続が必要です。

問題が起きるところ

翻訳版のパスでアイコンがすべて404になる。 相対指定の `src` はマニフェストのURLを基準に解決されるため、言語の接頭パス配下で配信したマニフェストは、アイコンもその配下に要求します。アイコンのアドレスは、ホストのルートから、または完全なURLで指定します。

インストールしたアプリが別の言語で開く。 1つのマニフェストでは起動URLを1つしか指定できず、全員が同じホームページをインストールすることになります。言語ごとに `start_url` と `id` を持つマニフェストを配信し、各ページから対応するものにリンクします。

インストール済みのアプリが新しいマニフェストを無視する。 `id` がなければ起動URLが識別情報になるので、起動ページを移すとファイルの識別情報も変わります。その場合、ブラウザーはインストール済みのものとは別のアプリとして扱うべきことになります。最初のリリース前に `id` を設定し、それ以後は変えません。

ランチャーがロゴを切り抜く。 用途が `any` のアイコンは、プラットフォームの判断で配置または切り抜かれます。短辺の40%を半径とする中央の円に重要な部分をすべて収めた、 `maskable` アイコンを追加します。

マニフェストを編集しても何も変わらない。 HTML標準では、 `link` 要素のうち、 `rel` に `manifest` を含む最初のものしか使いません。そのため、テーマやプラグインが先に独自のリンクを入れていると、編集しているファイルは一度も読まれません。マニフェストへのリンクは、1ページに必ず1つだけにします。

参考資料

- Web Application Manifest — 全メンバー、ブラウザーによる処理、 `id` 、 `scope` 、アイコンの用途に関する規則。
- Web App Manifest - Application Information — `description` 、 `categories` 、 `screenshots` と、画面形状やラベル。
- IANA: application/manifest+json — メディアタイプと拡張子 `.webmanifest` の登録。
- HTML標準: manifestリンク種別 — マニフェストのリンクを取得する方法と、最初のリンクだけを使う理由。
- Windows版Firefoxでウェブアプリを使う — Windowsでの機能と、143以降、およびMicrosoft Storeからインストールする Firefoxでは150以降で利用できること。
- Firefox Source Docs: Firefoxのウェブアプリ — 現在の実装。Windowsでは初期状態で有効、Linuxでは無効、macOSでは未対応。

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

「ウェブアプリマニフェストとは？」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/webmanifest/