Engineer

Web-App-Manifest

JSON-Datei, die eine Site als installierbare App beschreibt

Was es ist

Ein Web-App-Manifest ist ein JSON-Dokument, das eine Website als Anwendung beschreibt: ihren Namen, ihre Startadresse, die zugehörigen Adressen, das Aussehen ihres Fensters und die Bilder, die sie repräsentieren. Eine Seite verweist mit einem `link`-Element und `rel` gleich `manifest` auf die Datei. Ein Browser, der eine Installation anbietet, liest daraus, welche Eigenschaften die installierte Fassung haben soll. Die W3C-Spezifikation nennt es schlicht Manifest, ebenso wie der übliche Sprachgebrauch.

Die Datei ist gewöhnliches JSON. Ihr registrierter Medientyp ist `application/manifest+json`, ihre registrierte Endung `.webmanifest`. Die Spezifikation akzeptiert jedoch jeden JSON-Medientyp, und viele Websites liefern eine `manifest.json` aus. Alle Einträge sind optional. Ein kleines Manifest für eine erfundene Website mit Gezeitentabellen sieht so aus:

```
{
    "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"
        }
    ]
}
```

Drei Einträge bestimmen, was die App ist, im Unterschied zu ihrem Aussehen. `start_url` schlägt die Seite vor, mit der die installierte App startet. Der Browser kann sie ignorieren oder dem Nutzer erlauben, sie zu ändern. `scope` ist die Menge der Adressen innerhalb der App. Fehlt dieser Eintrag, verwendet die Spezifikation die Start-URL ohne Dateinamen, Abfrage und Fragment. Im Beispiel gehört damit alles unter `/tides/` zur App. `id` ist ihre Identität und fällt, wenn es fehlt, auf die Start-URL zurück. Relative Iconadressen und die Start-URL werden relativ zur URL des Manifests aufgelöst, nicht zu der Seite, die es verlinkt. Eine relative `id` bezieht sich dagegen auf die Origin der Start-URL.

Die übrigen Einträge betreffen die Darstellung. `display` fordert `fullscreen`, `standalone`, `minimal-ui` oder `browser` an. Kann ein Browser die gewünschte Darstellung nicht umsetzen, geht er diese Liste in der angegebenen Reihenfolge weiter. Jedes Icon kann einen `purpose` haben: standardmäßig `any`, `maskable` für ein Bild, das einen Beschnitt verträgt, oder `monochrome` für eines, dessen Farbe verworfen und nur dessen Alphakanal verwendet wird. `theme_color` färbt den Fensterrahmen der App. Eine ergänzende W3C-Note beschreibt `description`, `categories` und `screenshots`, die Installationsdialoge und App-Stores anzeigen können.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Web-App-Manifest?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

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. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Warum es wichtig ist

Das Manifest liefert Installationsmetadaten unter der Kontrolle der Website. Die Spezifikation überlässt die Installationsentscheidung dem Browser: Er kann einen Installationsweg anbieten und legt seine Bedingungen selbst fest. Einige davon verlangen bestimmte Einträge. Das Team liefert die Eingabe, der Browser entscheidet über ihre Verwendung. Die nützliche Frage ist daher weniger, ob ein Manifest nötig ist, sondern was es verspricht.

Die Identität ist das Versprechen, das die Datei überdauert. Ruft ein Browser das Manifest erneut ab, soll er laut Spezifikation eine übereinstimmende `id` als Hinweis verstehen, dass die neue Datei das Manifest einer bereits installierten App ersetzt. Ohne `id` ist die Start-URL die Identität. Wird die Startseite verschoben, ändert sich diese stillschweigend, und der Browser soll die neue Datei als andere Anwendung behandeln als die auf dem Gerät des Lesers. Eine feste `id` ab der ersten Veröffentlichung kostet eine Zeile und beseitigt dieses Risiko.

Der Geltungsbereich bestimmt, wo das App-Fenster endet. Navigation innerhalb dieses Bereichs bleibt im installierten Fenster. Ein Link nach außerhalb öffnet sich weiterhin, doch die Spezifikation erwartet, dass der Browser den Wechsel aus der App kenntlich macht. Ein engerer Bereich als die Website lässt damit normale Links zwischen ihren Abschnitten wie Ausgänge wirken.

Bei Icons entstehen die meisten sichtbaren Schäden. Startoberflächen schneiden sie zu Kreisen, abgerundeten Quadraten und anderen Formen zu. Ein Logo, das sein ganzes Quadrat ausfüllt, verliert dabei die Ränder. Für ein maskierbares Icon definiert die Spezifikation die sichere Zone als mittigen Kreis mit einem Radius von zwei Fünfteln der kürzeren Iconseite. Alles außerhalb darf abgeschnitten werden. Ein monochromes Icon behält nur seinen Alphakanal. Ein Icon auf einem undurchsichtigen Hintergrund wird dadurch zu einem ausgefüllten Quadrat.

Die Farbeinträge haben einen engeren Zweck, als es scheint. `background_color` ist nur für den Moment gedacht, bevor das Stylesheet eintrifft. Die Spezifikation verbietet dem Browser, den Wert zu verwenden, sobald das Stylesheet verfügbar ist. Der Eintrag bestimmt nur die Hintergrundfarbe der ersten Darstellung.

Wie es in der Praxis aussieht

Diese Website erzeugt ihr Manifest aus einer Vorlage, statt eine Datei von Hand zu pflegen. Die Hugo-Konfiguration registriert `application/manifest+json` mit der Endung `webmanifest` und legt das Manifest als Ausgabe der Startseite jeder Sprache fest. So bekommt jede Sprache eine eigene Datei, und der Kopf jeder Seite verlinkt die Datei ihrer Sprache. Die Konfiguration hält den Grund fest: Eine globale Fassung kann nur eine Start-URL nennen. Ein Besucher, der die dänische Fassung der Website installierte, bekam deshalb eine App mit der englischen Startseite.

Die Vorlage baut das Manifest als Zuordnung auf und übergibt sie Hugos JSON-Kodierer. Escapezeichen und Kommas sind damit dessen Aufgabe; ein fehlendes Komma kann nicht veröffentlicht werden. Jede Iconadresse beginnt an der Wurzel des Hosts. Der Kommentar begründet das ausdrücklich: Eine relative Adresse in `/da/manifest.webmanifest` würde unter `/da/` aufgelöst, und jedes Icon der dänischen Fassung würde einen 404-Fehler liefern. Die Icons selbst werden aus drei Ordnern im Dateisystem gelesen, einem pro Verwendungszweck, mit je zehn Größen von 36 bis 512 Pixeln. Die Größe stammt aus dem Dateinamen. Eine vom Generator ergänzte Größe erscheint daher ohne weitere Änderung im Manifest.

`id` ist auf die Startadresse der jeweiligen Sprache festgelegt, `scope` umfasst die gesamte Origin. Ein Link in eine andere Sprache bleibt somit im installierten Fenster. Screenshots heißen `<lang>-<form_factor>-<WxH>.webp`. Die Vorlage liest alle drei Angaben aus dem Dateinamen: Englisch und Dänisch haben jeweils eine breite Aufnahme mit 1.280 mal 800 und eine schmale mit 780 mal 1.688 Pixeln. Bei einer Sprache ohne Screenshots fehlt der Eintrag `screenshots` vollständig, statt eine leere Liste zu enthalten.

Die Credits-Seite führt die W3C-Spezifikation unter den Standards der Website auf und PWABuilder unter den externen Prüfungen, die jeder ausführen kann. Der Dienst liest von Microsofts Rechnern aus das Manifest und die übrigen Bestandteile, die die Website installierbar machen.

Die Installationsseite erklärt dieselbe Datei aus Lesersicht. Sie zeigt, wo die einzelnen Browser den Installationsbefehl anbieten, und weist darauf hin, dass Apples Browser nie von sich aus dazu auffordern. Firefox bietet ein separates Fenster für Web-Apps, unter Windows standardmäßig ab Version 143 beziehungsweise bei Installationen aus dem Microsoft Store ab Version 150. Mozillas Implementierungsdokumentation nennt Linux-Unterstützung als standardmäßig deaktiviert und macOS-Unterstützung als nicht verfügbar. Die Seite klärt auch, was bei dieser Website zu erwarten ist: Die installierte App ist ein Icon und ein eigenes Fenster. Die Website veröffentlicht weder einen Service Worker noch eine verwaltete Offlinekopie. Für die normale Nutzung braucht sie daher eine Verbindung.

Wo es schiefgeht

Alle Icons liefern unter übersetzten Pfaden 404. Ein relatives `src` wird relativ zur URL des Manifests aufgelöst. Ein Manifest unter einem Sprachpräfix fordert Icons deshalb unter diesem Präfix an. Schreiben Sie Iconadressen von der Wurzel des Hosts aus oder als vollständige URLs.

Die installierte App startet in der falschen Sprache. Ein Manifest kann eine Start-URL angeben, also installieren alle Leser dieselbe Startseite. Liefern Sie für jede Sprache ein eigenes Manifest mit eigener `start_url` und `id` aus und verlinken Sie auf jeder Seite die passende Fassung.

Installierte Fassungen ignorieren das neue Manifest. Ohne `id` bestimmt die Start-URL die Identität. Das Verschieben der Startseite gibt der Datei daher eine neue Identität, und der Browser soll sie als andere App behandeln als die bereits installierte. Legen Sie `id` vor der ersten Veröffentlichung fest und lassen Sie den Wert danach unverändert.

Die Startoberfläche beschneidet das Logo. Ein Icon mit dem Zweck `any` wird nach Ermessen der Plattform platziert oder zugeschnitten. Ergänzen Sie ein `maskable`-Icon, bei dem alles Wichtige innerhalb des mittigen Kreises liegt, dessen Radius 40 % der kürzeren Iconseite beträgt.

Änderungen am Manifest bleiben ohne Wirkung. Der HTML-Standard verwendet nur das erste `link`-Element, dessen `rel` den Wert `manifest` enthält. Hat ein Theme oder Plugin vorher einen eigenen Link eingefügt, wird die bearbeitete Datei gar nicht gelesen. Belassen Sie auf jeder Seite genau einen Manifest-Link.

Weiterführende Lektüre

- Web Application Manifest — alle Einträge, ihre Verarbeitung durch den Browser und die Regeln für `id`, `scope` und Iconzwecke.
- Web App Manifest - Application Information — `description`, `categories` und `screenshots` mit Formfaktoren und Beschriftungen.
- IANA: application/manifest+json — Registrierung des Medientyps und der Endung `.webmanifest`.
- HTML Standard: link type manifest — Abruf des Manifest-Links und der Grund, warum nur der erste zählt.

- Use web apps in Firefox for Windows — die Windows-Funktion ab Version 143 beziehungsweise im Microsoft Store ab 150.
- Firefox Source Docs: Web Apps in Firefox — die aktuelle Implementierung: unter Windows standardmäßig aktiviert, unter Linux standardmäßig deaktiviert, unter macOS nicht unterstützt.

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 ein Web-App-Manifest?“ 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/webmanifest/