PWA – Progressive Web Apps

What

- Web Applications built with modern Web APIs.

    - reliable
        - gives a good experience regardless of network speed
    - installable
        - users can install the app and open in a standlaone window
    - linkable
        - app is linkable and discoverable without the need for an app store
    - universal
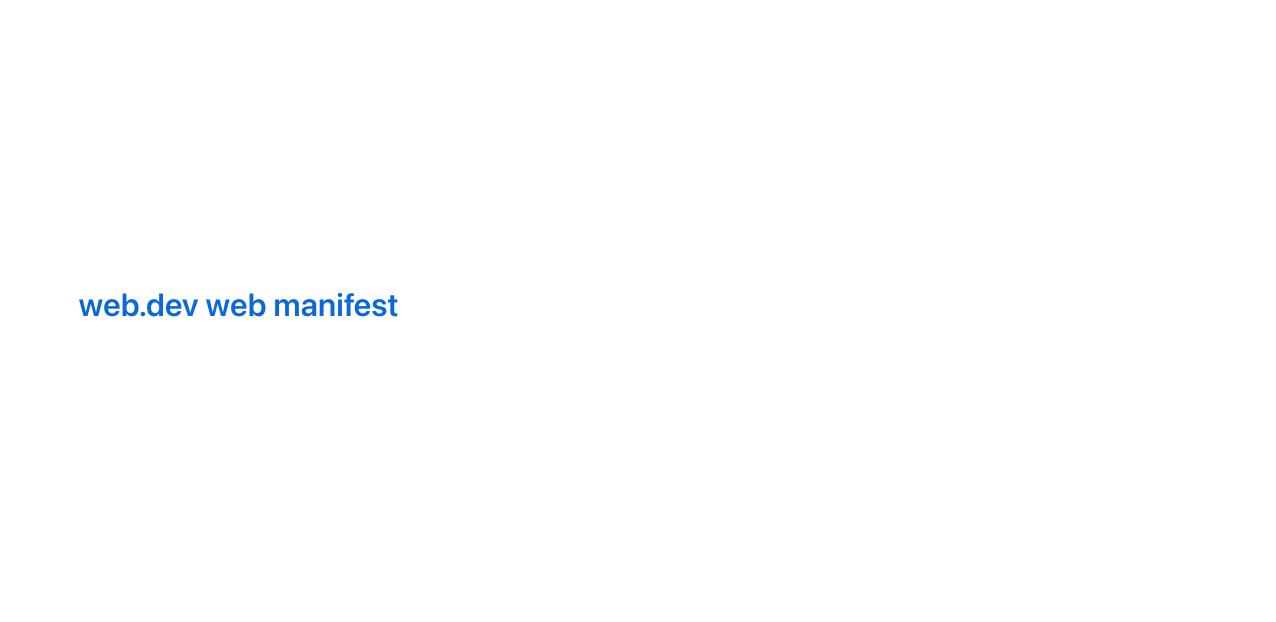   - one codebase works in any device

How

- Web App Manifest
- Web APIs
  - Service Worker API
  - Cache
- Secure
  - Uses HTTPS
  - Has a valid SSL Certificate
  - No mixed content (e.g. content loaded via HTTP)

Contains the necessary information for the web app to be downloaded and be presented to the user similarly to a native app.

```html
<link rel="manifest" href="manifest.json" />
```

```json
{
  "$schema": "https://json.schemastore.org/web-manifest-combined.json",
  "name": "HackerWeb",
  "short_name": "HackerWeb",
  "start_url": ".",
  "display": "standalone",
  "background_color": "#fff",
  "theme_color": "#C3C3C3",
  "description": "A readable Hacker News app.",
  "icons": [{
    "src": "images/touch/homescreen48.png",
    "sizes": "48x48",
    "type": "image/png"
  }, {
    "src": "images/touch/homescreen72.png",
    "sizes": "72x72",
    "type": "image/png"
  }, {
    "src": "images/touch/homescreen192.png",
    "sizes": "192x192",
    "type": "image/png"
  }],
```

web.dev web manifest

Service Workers

"Service workers essentially act as **proxy servers that sit between web applications, the browser, and the network (when available)**. They are intended, among other things, to enable the creation of effective offline experiences, intercept network requests and take appropriate action based on whether the network is available, and update assets residing on the server." – MDN

Service Worker Example

```javascript
self.addEventListener('install', event => {
  console.log('V1 installing…');

  // cache a cat SVG
  event.waitUntil(
    caches.open('static-v1').then(cache => cache.add('/dog.svg'))
  );
});

self.addEventListener('activate', event => {
  console.log('V1 now ready to handle fetches!');
});

self.addEventListener('fetch', event => {
  const url = new URL(event.request.url);

  // serve the cat SVG from the cache if the request is
  // same-origin and the path is '/dog.svg'
  if (url.origin == location.origin && url.pathname == '/cat.svg') {
    event.respondWith(caches.match('/dog.svg'));
  }
});
```

Installation

Prerequisites

App must:

- be served via HTTPS

- have at least one icon in the correct format and size (512 by 512 pixels)

- have a registered service worker
  - with a `fetch` event handler for basic offline experiences

Enhanced Capabilities

- Web APIs
  - Background Sync API
  - Push API
  - Fullscreen API
  - File System Access API
  - Contact Picker API
  - Notifications API
  - Vibration API
  - Battery API
  - Bluetooth API
  - Web NFC API
  - Web Share API (YIR)