Progressive Web Apps

Part 3. The Web App Manifest

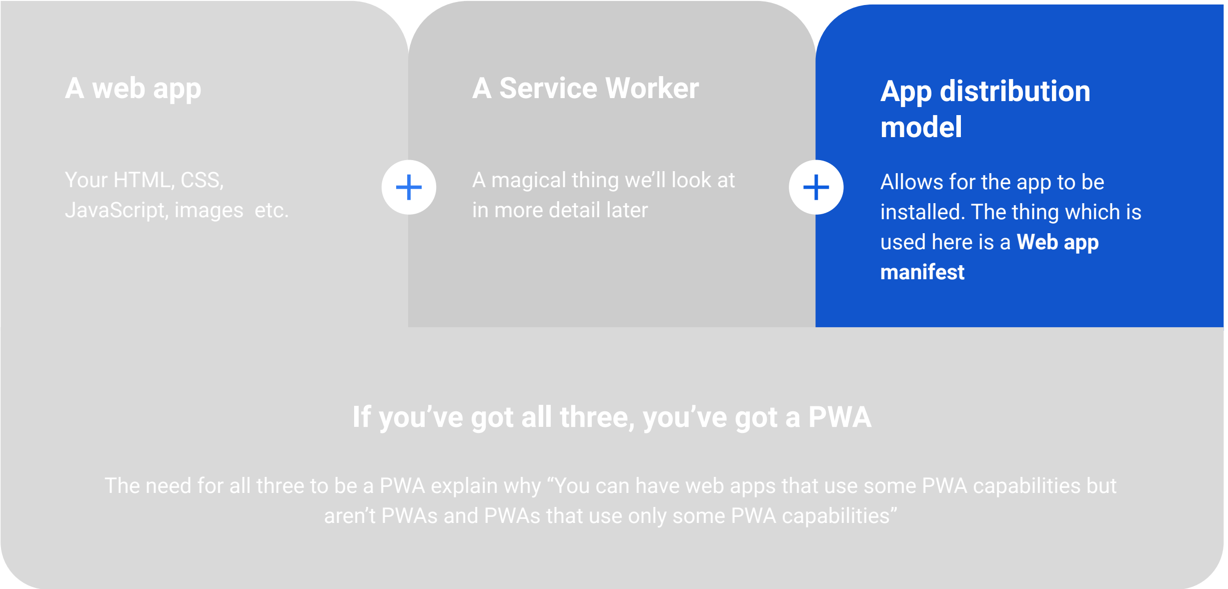
Recap

This is the last session we’ll be doing on PWA. In this series so far we’ve looked at:

1. An overview of PWA capabilities and associated technologies
2. The Service Worker in a fair amount of detail

We are now going to look at the app distribution model, specifically the Web App Manifest

> “the web app manifest is a PWA feature that is really mobile-focused. With it, you can **specify details about your app** that help devices give your users the best possible experience..”
>
> Beginning Progressive Web App Development, Dennis Sheppard (O’Reilly)



A web app

Your HTML, CSS, JavaScript, images etc.

+

A Service Worker

A magical thing we’ll look at in more detail later

+

App distribution model

Allows for the app to be installed. The thing which is used here is a **Web app manifest**

If you’ve got all three, you’ve got a PWA

The need for all three to be a PWA explain why “You can have web apps that use some PWA capabilities but aren’t PWAs and PWAs that use only some PWA capabilities”

The app distribution model

A bit of background

Before there were Web App Manifests…

We still needed to describe how a site should look if a user should add it to their home screen. This was pretty basic, focussed mostly on icons, but the implementation was proprietary.

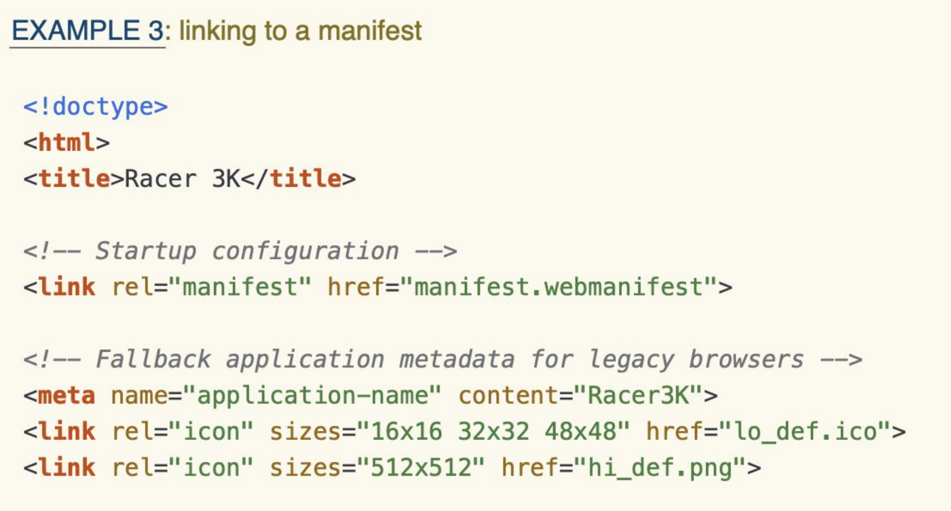
*"When Apple released the iPhone, it added a handful of proprietary meta tags to describe what a web app should look like, what icon should be displayed, and how it would behave if someone added it to their homescreen. Because these meta tags were specific to Apple devices, Google and Microsoft eventually added their own meta tags. If a web developer supported all three platforms, the resulting HTML document would be littered with **nearly two dozen meta tags.**"* Progressive Web App, Jason Grigsby (A Book Apart)

You can see some of these here:
https://thishereweb.com/understanding-the-manifest-for-web-app-3f6cd2b853d6

Now you simply link to a manifest

> “We need to make sure that we’ve provided the browser with t**he information necessary to support the various ways that users might add an icon to their home screen**. The primary way we do this is through the manifest file”

Progressive Web App, Jason Grigsby (A Book Apart)



EXAMPLE 3: linking to a manifest

```
<!doctype>
<html>
<title>Racer 3K</title>

<!-- Startup configuration -->
<link rel="manifest" href="manifest.webmanifest">

<!-- Fallback application metadata for legacy browsers -->
<meta name="application-name" content="Racer3K">
<link rel="icon" sizes="16x16 32x32 48x48" href="lo_def.ico">
<link rel="icon" sizes="512x512" href="hi_def.png">
```

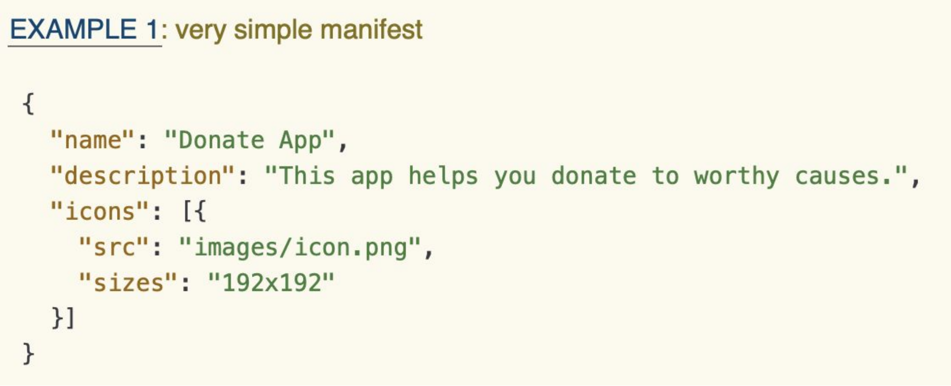
See more detail at:
https://w3c.github.io/manifest/#using-a-link-element-to-link-to-a-manifest

A simple manifest file

Manifests are simple JSON files where the properties are referred to as ‘members’.

The official extension is .webmanifest, but you’ll also see people using simply .json

Note: W3C strongly suggest you transfer the manifest using the application/manifest+json MIME type.



EXAMPLE 1: very simple manifest

```
{
  "name": "Donate App",
  "description": "This app helps you donate to worthy causes.",
  "icons": [{
    "src": "images/icon.png",
    "sizes": "192x192"
  }]
}
```

See more detail at:
https://w3c.github.io/manifest/#example-manifests

Other things a manifest can specify

The Web App Manifest provides a consistent means of specifying many things. See a more complete list here: https://developer.mozilla.org/en-US/docs/Web/Manifest

- **Categories**: “education”, “utilities”, “news” etc.
- **Description**
- **Display**: “standalone”, “minimal UI”, “fullscreen”, “browser” etc.
- **IARC rating ID** (International Age Rating Coalition)*
- **Icons, Name** (which will be shown on a home screen)
- **Scope** (of the application’s context)
- **Screenshots** (for stores)
- **Start URL**

* As an aside, IARC is quite interesting: https://www.globalratings.com/how-iarc-works.aspx

Installability signals

There's no "install API" provided by the manifest specification

This is important to remember. We referred to this in the first session as a browser's installability criteria needing to be met before a user will be prompted by to install your PWA.

The spec refers to the manifest as servings as an ***installability signal*** to the user agent that app can be installed. As we discussed, there are several others too. Essentially, it's up to the user agent:

- If they consider the app installable
- How they'll notify the user of this