Mads Kristensen

A blog about ASP.NET & Visual Studio

Progressive Web Apps made easy with ASP.NET Core

Published Nov 15, 2017

Earlier this week I wrote about some experiments I was doing with Service Workers <https://developers.google.com/web/fundamentals/primers/service-workers/> in ASP.NET Core. This is an update to that.

So, what is a Progressive Web App (PWA)?

> *A Progressive Web App uses modern web capabilities to deliver an app-like user experience – Progressive Web Apps <https://developers.google.com/web/progressive-web-apps/>*

The benefits of PWAs are many. My personal favorites include:

» They are faster than regular websites

- They are more reliable
- They work offline
- They can be installed on the desktop or phone
- Most major browsers already support them (Safari and Edge coming soon)

Any website or web application can add the capabilities that turns them into PWAs. The capabilities to add are:

1. The website must be served over HTTPS
2. Add a Web App Manifest (it's a simple JSON file)
3. Add a Service Worker (a JavaScript file)

With ASP.NET Core we can automate a lot of this to make it easier and more integrated with the rest of the application.

Getting started

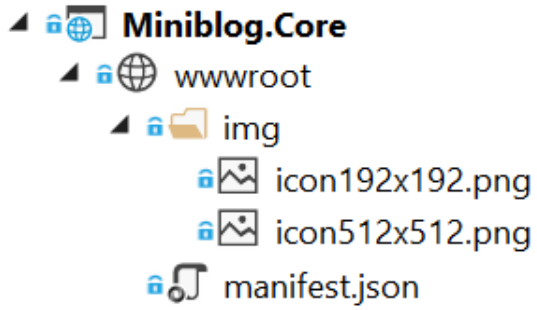
So, let's get started turning our ASP.NET Core web application into a full fledged PWA by following a few easy steps. This will make your site work offline, be faster and installable by supporting browsers.

Step 1 – install a NuGet package

Install the NuGet package [WebEssentials.AspNetCore.PWA <https://www.nuget.org/packages/WebEssentials.AspNetCore.PWA/>](https://www.nuget.org/packages/WebEssentials.AspNetCore.PWA/) into your ASP.NET Core project.

Step 2 – add a manifest and icons

Add a file called *manifest.json* in the *wwwroot* folder as well as 2 image icons.



You can have as many image icons as you want as long as you have one in the size of 192x192 and one 512x512 pixels.

Fill in the *manifest.json* file similar to this:

```
{
  "name": "Awesome Application",
  "short_name": "Awesome",
  "description": "The most awesome application in the world",
  "icons": [
    {
      "src": "/img/icon192x192.png",
      "sizes": "192x192"
    },
    {
      "src": "/img/icon512x512.png",
      "sizes": "512x512"
    }
  ],
  "display": "standalone",
  "start_url": "/"
}
```

Step 3 – register a service

Inside the `ConfigureServices` method in *Startup.cs*, add a call to `services.AddProgressiveWebApp()` like so:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();
    services.AddProgressiveWebApp();
}
```

Voila! The app is now a full blown PWA.

To verify it works and your web app now behaves like a PWA in supported browsers, check out the [verification step on the project readme <https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker#step-4---verify-it-works>](https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker#step-4---verify-it-works) .

Also see how it is implemented in the [Miniblog.Core source code <https://github.com/madskristensen/Miniblog.Core>](https://github.com/madskristensen/Miniblog.Core) which is a production web app. This very website (madskristensen.net) is also running it.

Configuration

There are plenty of ways to configure the behavior of the Web App Manifest as well as the service worker. Read the [documentation <https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker>](https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker) for more info.

I do want to call out that the NuGet package creates a strongly typed object (*WebEssentials.AspNetCore.Pwa.WebManifest*) out of the *manifest.json* file and makes it available in the dependency injection system. That way you can have a single source of truth for meta data properties such as application name, description and icon list.

Next steps

Read the full description <https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker> of using the NuGet package where you'll also find links to the various specifications and videos about PWAs.

Contribute

If this is of interest to you and you'd like to contribute to the project on GitHub <https://github.com/madskristensen/WebEssentials.AspNetCore.ServiceWorker> then you are more than welcome to do so. Open bugs, suggest features and send pull requests are all appreciated.