

# PWA using Angular 7.2 Service-Worker

## PWA Ideas, AppShell, Manifest, Service Worker spec, CLI's --service-worker, Implementation, Project

Web developers have been enviously glancing over at native app developers and admiring all the shiny extra toys they have – the sometimes connected app, instant app start, app store for discoverability, notifications, etc. (of course, web developers do not forget the unique advantages they have – a ubiquitous platform, use of URLs for deep linking, avoiding version hell, etc.).

With the arrival of Progressive Web Applications (PWA), now web developers have the best of both worlds. PWA makes it easy to build web apps that run in modern browsers and behave like native apps.

A PWA is a web app that uses three key technologies – the manifest, service workers and optionally the app shell. The best way to build PWAs is to use Angular 7 & its *Service-Worker* package. The project source tree can be created as normal using Angular 7.2 CLI and then run this to add PWA support: `ng add @angular/pwa`

The aim of this course to to bring Angular 7.2 developers up to speed with how a PWA works, to review the underlying spec and then to dive deeply into how to implement a powerful PWA using Angular's Service-Worker package.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> Web developers wishing to build powerful PWA apps using Angular 7.2 and its <i>Service-Worker</i> package.</p> <p><b>Prerequisites</b> Good experience of Angular 7.2 programming.</p> <p>This is an advanced course and before attending, attendees should already be quite familiar with Angular 7.2 and TypeScript programming.</p>	<p style="text-align: center;"><b>PWA – What are we trying to achieve</b></p> <p>All three ideas are important:</p> <ul style="list-style-type: none"> <li>- progressive</li> <li>- web</li> <li>- applications</li> </ul> <p>The idea of a client-side mini-proxy server and transient network connections</p> <p>The fact we gain native app capabilities does not mean we lose web app capabilities</p> <p style="text-align: center;"><b>PWA Technologies</b></p> <p>Manifest Service Worker AppShell</p> <p>Review of how these technologies work</p> <p>Message flows</p> <p>Service workers are quite distinct from web workers – not to be confused</p> <p style="text-align: center;"><b>Intro to PWA with Angular 7.2</b></p> <p>Angular and Progressive Web Apps</p> <p>What Angular 7.2 offers to PWA app devs</p> <p>Exploring the Service-Worker package</p> <p>Angular CLI and Service-Worker apps</p> <p style="text-align: center;"><b>AppShell</b></p> <p>An approach to initially offer a minimalist UI that can be cached</p> <p>Gradually add more content</p> <p>Very fast rendering of first view</p> <p style="text-align: center;"><b>Manifest</b></p> <p>What's in a manifest.json file?</p> <p>Generating default</p> <p>Start url, scope, display, etc.</p>
	<p style="text-align: center;"><b>Service Workers Spec</b></p> <p>Review of <a href="#">W3C Service Workers Spec</a> – we explore how service workers perform</p> <p>Main artifacts</p> <p style="text-align: center;"><b>Angular CLI Generated Code</b></p> <p>Add PWA using: <code>ng add @angular/pwa</code></p> <p>What does it do?</p> <p>Add new packages: ServiceWorker &amp; PWA</p> <p><code>angular.json/configurations/serviceWorker</code></p> <p><code>ServiceWorkerModule.register</code> call</p> <p style="text-align: center;"><b>ngsw-config.json</b></p> <p><code>assetGroups</code> <code>InstallMode</code> (prefetch) <code>UpdateMode</code> Resources</p> <p style="text-align: center;"><b>Working With Service Workers</b></p> <p><code>swUpdate</code>: deciding on an update strategy</p> <p><code>swPush</code>: service worker's push notifications</p> <p>Other aspects of service worker programming and configuration</p> <p style="text-align: center;"><b>Tooling</b></p> <p><a href="#">Google's PWA site</a> <a href="#">Google LightHouse</a></p> <p>Debugging and instrumentation</p> <p style="text-align: center;"><b>Internals</b></p> <p>The source for <a href="#">Angular's Service-Worker</a> package is well worth studying</p> <p>We also look at source for Angular CLI's <code>@angular/pwa</code> package</p> <p style="text-align: center;"><b>Project</b></p> <p>Creating an Angular 7.2 application that builds on the ideas explored in this course and that scores 100 in LightHouse</p>