PWA using Angular 8 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 8 & its *Service-Worker* package. The project source tree can be created as normal using Angular 8 CLI and then run this to add PWA support: `ng add @angular/pwa`

The aim of this course to to bring Angular 8 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.

Target Audience

Web developers wishing to build powerful PWA apps using Angular 8 and its *Service-Worker* package.

Prerequisites

Good experience of Angular 8 programming.

This is an advanced course and before attending, attendees should already be quite familiar with Angular 8 and TypeScript programming.

Contents of One-Day Training Course

PWA – What are we trying to achieve

All three ideas are important:
- progressive
- web
- applications

The idea of a client-side mini-proxy server and transient network connections
The fact we gain native app capabilities does not mean we lose web app capabilities

PWA Technologies

Manifest
Service Worker
AppShell
Review of how these technologies work
Message flows
Service workers are quite distinct from web workers – not to be confused

Intro to PWA with Angular 8

Angular and Progressive Web Apps
What Angular 8 offers to PWA app devs
Exploring the Service-Worker package
Angular CLI and Service-Worker apps

AppShell

An approach to initially offer a minimalist UI that can be cached
Gradually add more content
Very fast rendering of first view

Manifest

What's in a manifest.json file?
Generating default
Start url, scope, display, etc.

Service Workers Spec

Review of W3C Service Workers Spec – we explore how service workers perform
Main artifacts

Angular CLI Generated Code

Add PWA using: `ng add @angular/pwa`
What does it do?
Add new packages: ServiceWorker & PWA
angular.json/configurations/serviceWorker
ServiceWorkerModule.register call

ngsw-config.json

assetGroups
InstallMode (prefetch)
UpdateMode
Resources

Working With Service Workers

`swUpdate`: deciding on an update strategy
`swPush`: service worker's push notifications
Other aspects of service worker programming and configuration

Tooling

Google's PWA site
Google LightHouse
Debugging and instrumentation

Internals

The source for Angular's Service-Worker package is well worth studying
We also look at source for Angular CLI's @angular/pwa package

Project

Creating an Angular 8 application that builds on the ideas explored in this course and that scores 100 in LightHouse