


Angular Advanced - Creating PWAs






Peter Kassenaar –
info@kassenaar.com



*Goal – Creating a Progressive
Web Application (PWA) With
Angular*

OfficeWindowsSurfaceXboxDealsSupportMore






Windows BlogsWindows 10DevicesWindows DeveloperMicrosoft Edge Developer

Developer Resources  Platform Changelog  Demos 

FEBRUARY 6, 2018 9:00 AM

Welcoming Progressive Web Apps to Microsoft Edge and Windows 10

By [Kyle Pflug](#), [Kirupa Chinnathambi](#), [Aaron Gustafson](#) and [Iqbal Shahid](#)

 SHARE  TWEET  SHARE  SHARE  SKYPE

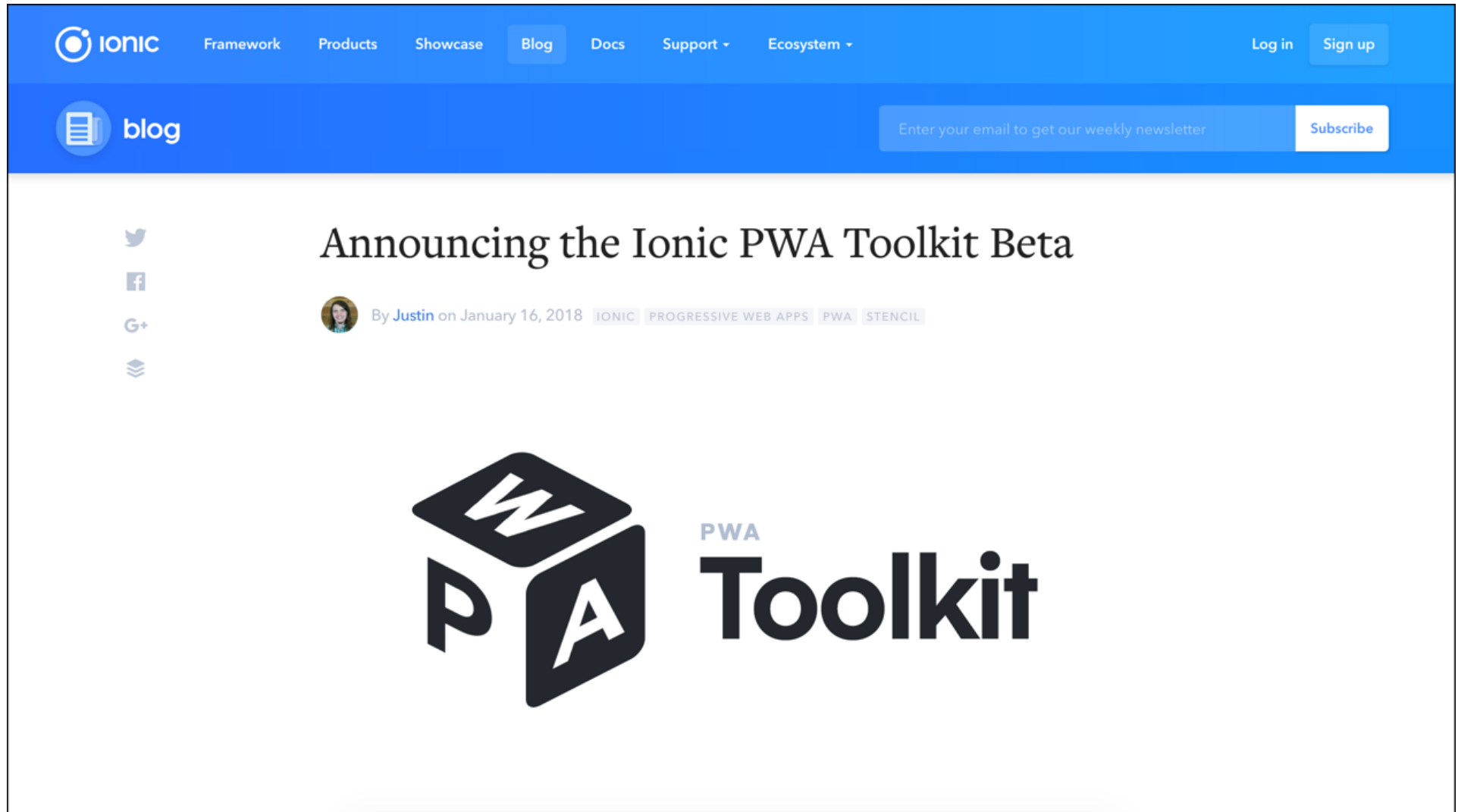
A little over a year ago, [we outlined our vision](#) to bring Progressive Web Apps (PWAs) to the more than half a billion devices running Windows 10. We believe PWAs are key to the web's future, and couldn't be more excited about their potential to enable more immersive web app experiences across all device

<https://blogs.windows.com/msedgedev/2018/02/06/welcoming-progressive-web-apps-edge-windows-10/>

Decisions – Manually or Framework?

- <https://blog.ionicframework.com/announcing-the-ionic-pwa-toolkit-beta/>
- <https://angularfirebase.com/lessons/hnpwa-angular-5-progressive-web-app-service-worker-tutorial/>

Framework?



<https://blog.ionicframework.com/announcing-the-ionic-pwa-toolkit-beta/>

When doing PWA with a framework

- Pro

- Quickly up and running
- Complete solution: routing, code splitting, pre-rendering, etc
- Possible extra's: icons, push notification, integration with native API's
- ...

- Con

- Possible overkill, bloat
- Less flexible
- You have to learn the framework first
- ...

When doing PWA manually

- Pro
 - Code all the details yourself
 - Full control
 - smaller, flexible apps
 - ...
- Con
 - Generally more work
 - Not easy upgrading
 - ...

What is needed to create a PWA?

Service Worker

“A *worker* is just a JavaScript file that sits in the background and does things that make web apps behave more like native mobile apps”

A service worker provides:

- cache static assets (making your app offline capable)
- cache network API calls
- trigger an install banner (on supported devices)
- send push notifications
- make your app faster for the end user
- **However:**
 - Not supported on Safari or Edge –
as of January 2018 (will be in the future)



Valid JSON Manifest file

- Configuration of the PWA
- Helps the service worker to create it's functionality

“Each time a visitor loads a new page, the service worker will cache the JSON extract and store it physically in the shell app.”

Additional stuff

- Add to Home Screen functionality,
- Icons in different sizes, resolutions,
- Mandatory: `https://` -connection
- More...

Creating a PWA with Angular-CLI

- Install Angular CLI >1.6
 - `npm install -g @angular/cli`
- Generate an app with Service Worker and routing:

```
ng new myPWA --service-worker --routing
```

Differences with non-PWA app

- In `.angular-cli.json`:
 - Option `serviceWorker` set to `true`
- This tells Angular to create a service worker file with your production build when you run `ng build --prod`

```
"apps": [  
  {  
    ...  
  },  
  "serviceWorker": true  
],
```

- In `app.module.ts`
- The `ServiceWorkerModule` needs to register the worker in the `app.module`

```
@NgModule({  
  ...  
  imports: [  
    ...  
    ServiceWorkerModule.register('/ngsw-worker.js',  
      { enabled: environment.production }  
    ),  
  ],  
  ...  
})
```

- New file: `ngsw-config.json` – auto generated for you
- One of the most important parts of the service worker!
- it tells Angular exactly *how to generate* the worker.
- More info: <https://medium.com/google-developer-experts/a-new-angular-service-worker-creating-automatic-progressive-web-apps-part-2-practice-3221471269a1>

Important groups

- `assetsGroup`
 - array that defines the static assets that are going to be cached by the service worker
- `dataGroups` :
 - Array of objects that holds information coming from external API's
 - Define what external content is cached and how it is refreshed
 - Various strategies for immediate refresh, or after cache invalidates (for instance after 4hr).

4. Create NGSW configuration file (default name is `src/ngsw-config.json`).

Here is the default content will be generated by Angular CLI 1.6 (you can find more details on the format of this configuration file in the [previous article](#)):

```
1  {
2    "index": "/index.html",
3    "assetGroups": [{
4      "name": "app",
5      "installMode": "prefetch",
6      "resources": {
7        "files": [
8          "/favicon.ico",
9          "/index.html"
10       ],
11      "versionedFiles": [
12        "/*_bundle.js"
```

<https://medium.com/google-developer-experts/a-new-angular-service-worker-creating-automatic-progressive-web-apps-part-2-practice-3221471269a1>

Default ngsw-config.json

```
{
  "index": "/index.html",
  "assetGroups": [{
    "name": "app",
    "installMode": "prefetch",
    "resources": {
      "files": [
        "/favicon.ico",
        "/index.html"
      ],
      "versionedFiles": [
        "/*.bundle.css",
        "/*.bundle.js",
        "/*.chunk.js"
      ]
    }
  }, {
    "name": "assets",
    "installMode": "lazy",
    "updateMode": "prefetch",
    "resources": {
      "files": [
        "/assets/**"
      ]
    }
  }
]}
```

Manifest.json

- You will need to create it manually.
- It contains some details about your app (icon, etc).
- Bundle it with your app, by adding it to `.angular-cli.json`!

```
"apps": [  
  {  
    "root": "src",  
    ...,  
    "assets": [  
      "manifest.json"  
    ],  
    ...  
  }  
]
```

Example manifest.json

```
{
  "short_name": "Cities-PWA",
  "name": "Angular 5.0 Cities PWA",
  "start_url": "/",
  "theme_color": "#f48c5b",
  "background_color": "#ffffff",
  "display": "standalone",
  "orientation": "portrait",
  "icons": [
    {
      "src": "/assets/icons/cities-icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

- Make sure to add at least one icon with a minimum of 512x512 pixels (you can add more to the array)
- More info on the manifest file: <https://developer.mozilla.org/en-US/docs/Web/Manifest>

Edit index.html to optimize for PWA

- Add `icon` link
- Add `manifest` link
- Provide a Splash page (i.e. loading page)
- Provide a `<noscript>`-section, if JavaScript disabled

`head>`

```
<meta charset="utf-8">
```

```
<title>Angular 5.0 Cities PWA</title>
```

```
<base href="/">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="icon" type="image/x-icon" href="/assets/icons/cities-icon-512x512.png">
```

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

```
<meta name="mobile-web-app-capable" content="yes">
```

```
<meta name="theme-color" content="#f48c5b">
```

```
<!-- add additional meta tags and stuff -->
```

`</head>`

Build the app

- `ng build --prod`
- Look inside the `/dist` folder of your app
- Serve the app, perform an Audit inside Chrome to see how it scores.
- Use the tab Application to (re)set or unregister Serviceworker

The image displays two side-by-side screenshots of the Chrome DevTools interface, specifically the 'Audits' and 'Application' tabs.

Left Screenshot (Audits Tab):

- The top bar shows the 'Audits' tab selected, with a red indicator for 4 errors.
- Four circular scores are displayed: Progressive Web App (82), Performance (43), Accessibility (100), and Best Practices (88).
- The 'Progressive Web App' section shows a score of 82 and lists 2 failed audits:
 - Does not redirect HTTP traffic to HTTPS (marked with a red X)
 - Content is not sized correctly for the viewport (marked with a red X, with a note: 'The viewport size is 521px, whereas the window size is 412px.')
- The 'Performance' section shows a score of 43.
- The 'Metrics' section shows a timeline of performance metrics with a bar chart and three red 'A' icons.

Right Screenshot (Application Tab):

- The top bar shows the 'Application' tab selected, with a red indicator for 4 errors.
- The 'Service Workers' section is expanded, showing two active service workers on localhost:
 - Service Worker 1:** Source `ngsw-worker.js`, Received 6-2-2018 14:44:59, Status #2102 activated and is running, Clients `http://localhost:5000/`.
 - Service Worker 2:** Source `ngsw-worker.js`, Received 6-2-2018 14:50:50, Status #2105 activated and is running, Clients `http://localhost:5000/`.
- Each service worker has 'Push' and 'Sync' buttons.



Workshop

- ...

More info

The image shows a YouTube video player interface. The video title is "Automatic Progressive Web Apps using Angular Service Worker – Maxim Salnikov – AngularConnect 2017". The video has 1,075 views. The video content shows a man speaking at a podium with the Angular logo and "INFRA" text in the background. To the right of the video is a social media feed for #angularconnect. The feed includes tweets from Katerina Skroumpelou (@skroumpelou), Matt Sharpe (@mattsharpe), and others. At the bottom right, there is a "Live stream sponsored by RANGLE.IO" banner. Below the video player, there is a "Volgende" (Next) section showing a preview of the next video, "Angular Animations: Motion Makes Meaning – Lukas".

YouTube NL Zoeken

ANGULAR CONNECT 2017

#angularconnect JOIN THE CONVERSATION

Getting ready for the first session in Saturn! #angularconnect @OCombe

Ready for some i18n at #angularconnect

We're sponsoring @Angular Come to the chill out room t or maybe join one of the

Live stream sponsored by RANGLE.IO

Automatic Progressive Web Apps using Angular Service Worker – Maxim Salnikov – AngularConnect 2017

1.075 weergaven

27 2 DELEN

Volgende

Angular Animations: Motion Makes Meaning – Lukas

<https://www.youtube.com/watch?v=wFjw0DM1ui4>

AngularFirestore

Search
PRO
Projects
Lessons
Snippets
Contact

You are receiving a free preview of 3 lessons

Contents

Angular 5 Service Worker - Everything you Wanted to Know

- Generating an App with the CLI >= 1.6
- angular-cli.json
- app.module.ts
- ngsw-config.json
- Asset Groups
- Data Groups - Cache External API Data
- manifest.json
- Building the App with a Service Worker

Let's build a Hacker News PWA

- Initial Setup
- index.html
- NgModules
- Data Service
- App Routing Module
- story-feed.component.ts
- story-item.component

Up Next - Performance Optimization

Recent Posts

Angular 5 Service Worker - Hacker News PWA Tutorial

Episode 72 Nov 22, 2017 by Jeff Delaney

Angular 5 Service Worker - Hacker News PWA

Angular 5 Service Worker

Angular HTTP

Angular Contentful

Angular v.5

0:03 / 7:40

It's a good time to be a web developer - **Progressive web apps** in Angular just got a whole lot easier as of November 2017. A few I created a PWA tutorial that used **SW-precache**, but today we can build a service worker automatically with Angular 5 by just se configuration options.

There is no better way to demonstrate a PWA than to build a **Hacker News** app and benchmark it against other frameworks.

Simplicity at Scale. Cloud infrastructure experience you'll love. \$100 Credit.

ads via Carbon

<https://angularfirebase.com/lessons/hnpwa-angular-5-progressive-web-app-service-worker-tutorial/>