Building Angular Progressive Web Apps!

Patrick Armentrout

About Me

- Working as a Software Engineer for ~ 8 years
- Started .Net MVC / jquery, knockout, etc..
- Angular + AngularJS Experience
    - Started with Angular 2.x rcs
- Tech Lead at Trimble Inc. - Westminster
- Enjoy Building Fun / Performant Web Applications

Let’s Talk About the Bachelor….

What is a Progressive Web App (PWA)?

Coined / invented by Google in 2015, a PWA is a set of WC3 standards that enable any web application to behave like a native app.

- **Progressive** to work for every user/browser
- **Responsive** / mobile first design
- "**Add to home screen**" feature allows for users to add to their app library
- Ability to work **offline** or over slow connections using **service workers**
- **Manifest file** to control the mobile user experience
  - Icons, splash screen, name, theming, etc..
- Ability to send **push notifications**
- Easily discoverable by **search engines**
  - Due to manifest and service worker registration scope
- Ability to interact with **phone hardware**

Browser / OS Support

Supported on all major Browsers:

- Chrome (40+)
- Firefox (44+)
- Edge (17+)
- Safari (11.1+)

Desktop Support

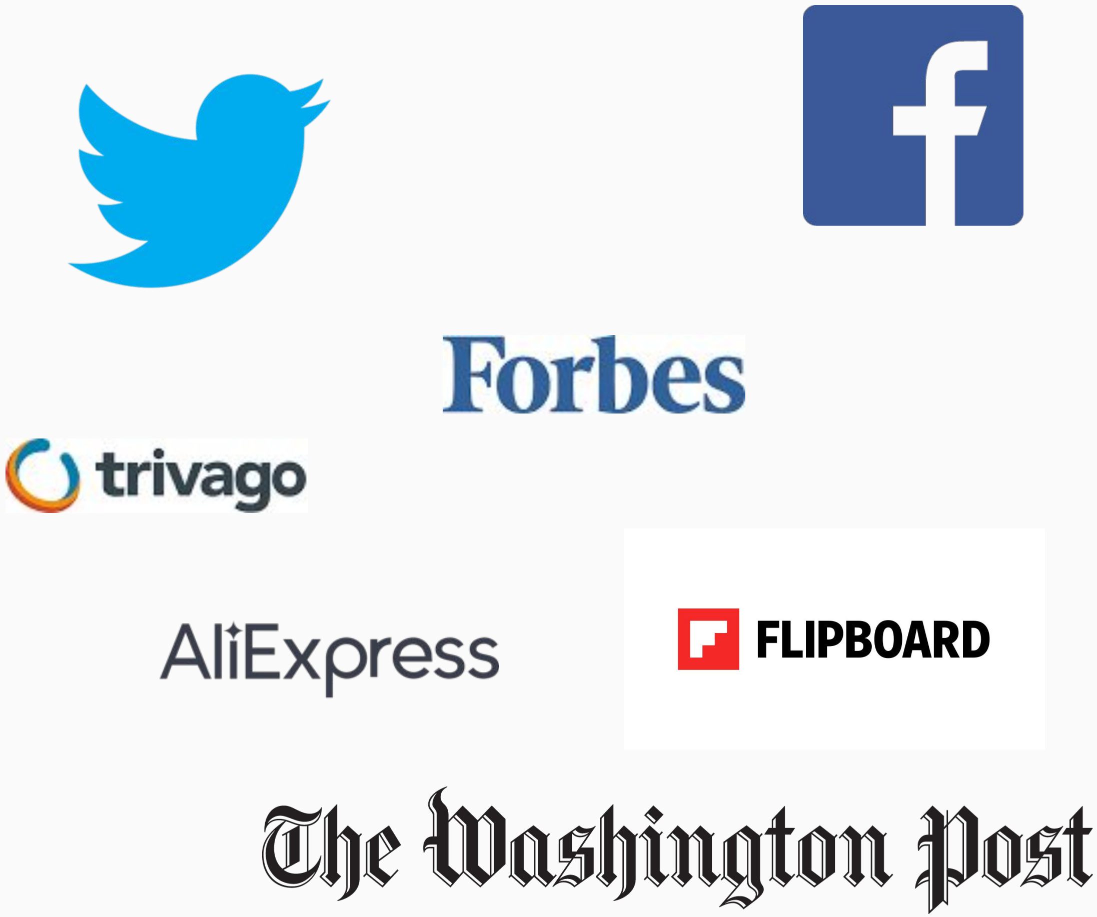
- Chrome OS (Chrome 67+)
- Linux (Chrome 70+)
- Windows (Chrome 70+)

For a deeper dive:

Service worker feature tracking

Google Developer Guide

Popular Progressive Web Apps



Angular PWA

```
ng new bachelor-app

ng add @angular/pwa --project bachelor-app
```

- Creates ngsw-config.json
- Creates manifest.json
- Adds Icons
- Updates index.html
- Updates app.module.ts
- Updates angular.json

Service Workers - Overview

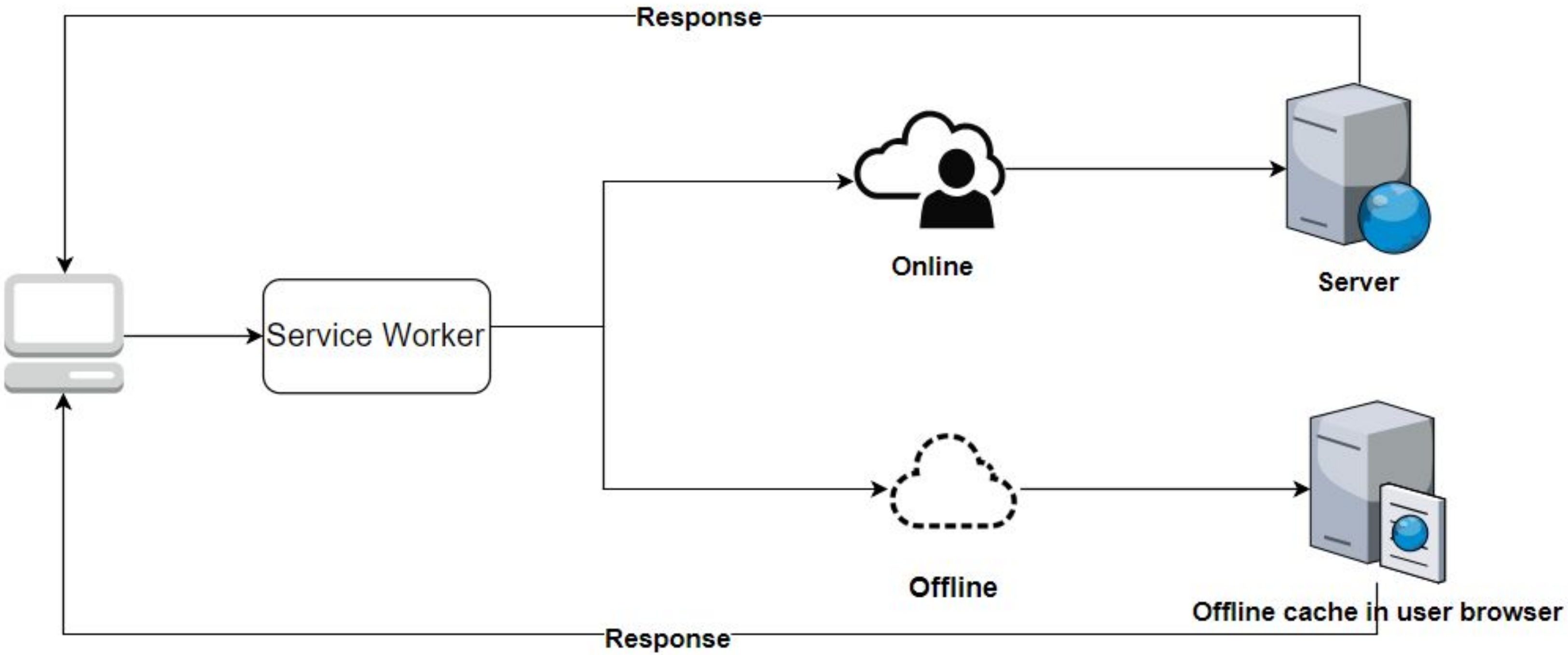


*Diagram via - https://altkomsoftware.pl

Service Worker Configuration - ngsw-config.json

Asset Groups

Caching for static website assets. (Html, css, js, image files, etc..)

Install / Update Mode

prefetch - tells the service worker to fetch and cache all listed resources (heavy up-front workload)

lazy - loads the resources as needed and caches accordingly

```
interface AssetGroup {
  name: string;
  installMode?: 'prefetch' | 'lazy';
  updateMode?: 'prefetch' | 'lazy';
  resources: {
    files?: string[];
    @deprecated
    versionedFiles?: string[];
    urls?: string[];
  };
}
```

Service Worker Configuration - ngsw-config.json (continued)

Data Groups

Defines custom caching for external data sources (i.e. api requests). For cacheConfig:

maxSize - maximum number of cached entries

maxAge - time until cache becomes invalidated

timeout - network timeout before service worker returns cached data

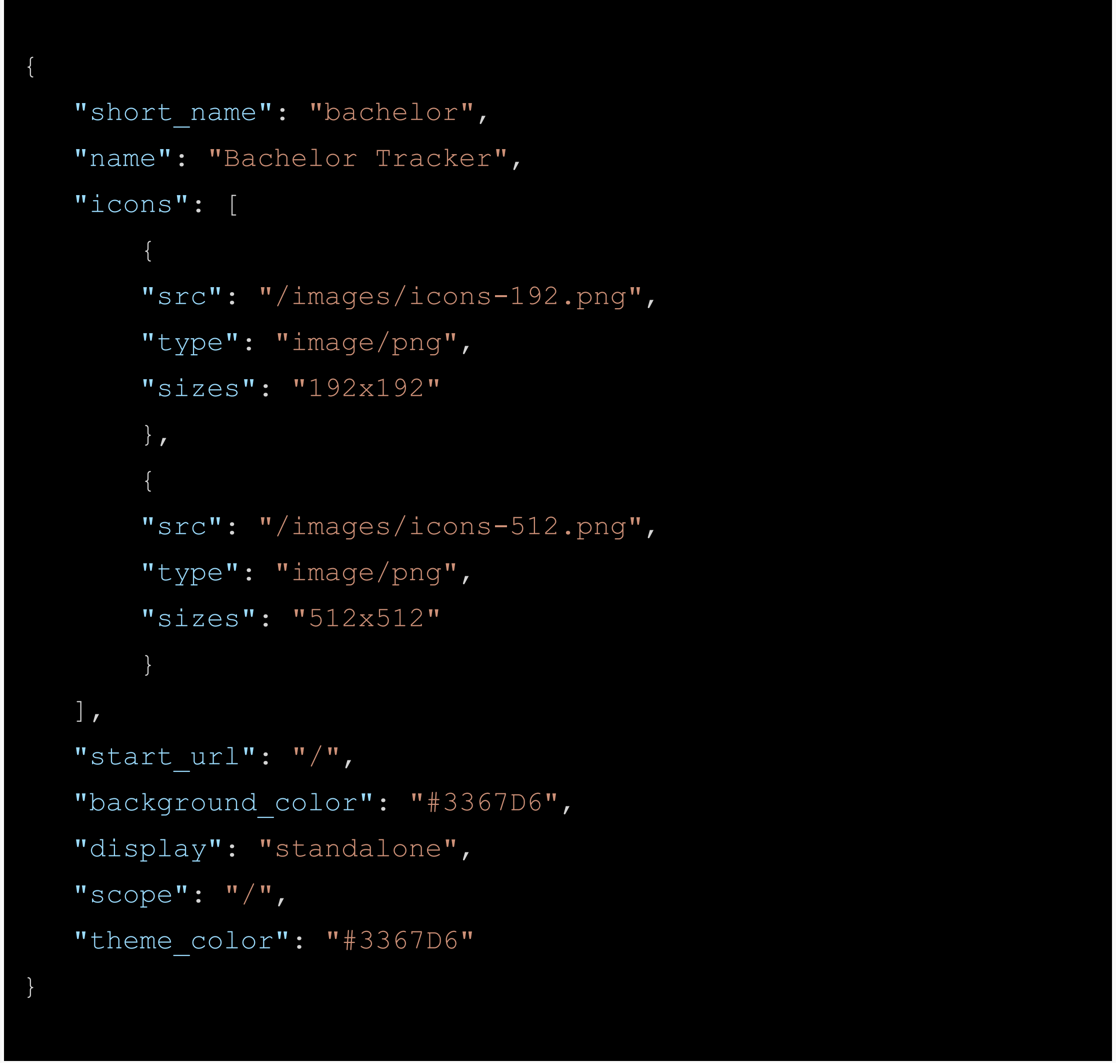
strategy - performance is “cache-first”, freshness is “network-first”

```
export interface DataGroup {
  name: string;
  urls: string[];
  version?: number;
  cacheConfig: {
    maxSize: number;
    maxAge: string;
    timeout?: string;
    strategy?: 'freshness' |
'performance';
  };
}
```

Manifest

Use the manifest.json file to customize the look and feel of your application!

Web Manifest Online Generator



```
{
    "short_name": "bachelor",
    "name": "Bachelor Tracker",
    "icons": [
        {
        "src": "/images/icons-192.png",
        "type": "image/png",
        "sizes": "192x192"
        },
        {
        "src": "/images/icons-512.png",
        "type": "image/png",
        "sizes": "512x512"
        }
    ],
    "start_url": "/",
    "background_color": "#3367D6",
    "display": "standalone",
    "scope": "/",
    "theme_color": "#3367D6"
}
```

Manifest Gotchas

- Theme color matches
  - index.html / manifest.json
- Icon sizes
  - 192px / 512px required
- Display
  - Standalone only for “add to home screen”
- Background color
  - Same as load page - smooth transition
- Orientation
  - Defaults to user device setting, only change if you must!

Mobile Demo!

Service Worker Module

By default - Angular cli will add the following to register your service worker to app.module.ts:

```
    ServiceWorkerModule.register('ngsw-worker.js', { enabled: environment.production }),
```

ServiceWorkerModule can also be used to check for updates:

```
swUpdate.available.subscribe(event => console.log(event.available));
```

Service Worker Module - Push Notifications!

Using SwPush service - prompt the user to subscribe for push notifications, and listen for messages to come in.

In Chrome, controlled by Firebase Cloud Messaging service. This is a proxy that we can use from the UI and Server to send messages to each user.

** Each browser uses a different messaging service so you may have to code accordingly

https://blog.angular-university.io/angular-push-notifications/

Chrome Lighthouse

Auditing Tool to test quality and performance of web pages.

Built in to Chrome Dev Tools, to go through PWA checklist.

https://developers.google.com/web/progressive-web-apps/checklist

Useful for reporting, testing, and can be integrated into a CI process

Thanks for coming!

Github:

https://github.com/Parmentrout/bachelor-pwa

Website:

https://bachelor-pwa.firebaseapp.com/

More PWA documentation / tutorials!

- https://developers.google.com/web/progressive-web-apps/
- https://angular.io/guide/service-worker-config
- PWA Tutorial 1
- PWA Tutorial 2