Automatic Progressive Web Apps using Angular Service Worker

Maxim Salnikov
Angular GDE
“How to create an Angular Progressive Web App?

Natively & naturally
Maxim Salnikov

- Full-Stack Engineer at ForgeRock
- PWAngelist / trainer
- LondonPWA and OsloPWA meetups organizer
- ngVikings conference organizer

@webmaxru
After all, what is PWA?

Progressive web apps use modern web APIs along with traditional progressive enhancement strategy to create cross-platform web applications.

These apps work everywhere and provide several features that give them the same user experience advantages as native apps.

https://developer.mozilla.org/en-US/Apps/Progressive
During last two weeks

[Updated] That’s official: after today’s release of iOS 11.3, #Safari joins the "#PWA in production" browsers hall of fame! It still lacks some features like Web Push & Background Sync, but @Apple’s decision to join the #YearOfPWA is clearly visible!

iOS 11.3 is available today

First PWAs have been published in the Windows 10 Store 🚀

- Student Doctor Network
- Skyscanner
- Oyster

PWAs installed via the store will appear in "app" contexts like Start and Cortana search results. 🙌

Another day, another platform supported for your #PWA: now these apps are native for Chrome OS. Offline, own window, launch icon, etc. Today on Canary channel, very soon in release. #YearOfPWA
UX advantages?

- Working offline
- Smarter networking
- Staying notified
- Proper app experience

Service Worker API

Web App Manifest
bit.ly/go-pw a-slack

• 1300+ developers
• Major browsers/ frameworks/libs reps
Angular Service Worker

NGSW
Generate a new Angular PWA

Yesterday

$ ng new myPWA --mobile

Angular Mobile Toolkit

- Angular CLI < 1.6
- Angular Service Worker β
Generate a new Angular PWA

Today

$ ng new myPWA --service-worker

- Angular CLI 1.6-1.7
- Angular Service Worker 5
Generate a new Angular PWA

Tomorrow

```
$ ng new myPWA
$ ng add @angular/pwa --project=myPWA
```

- Angular CLI 6
- Angular Service Worker 6
- Web App Manifest
- @Schematics
Building Angular PWA

```
$ ng build --prod

dist/
  ngsw-worker.js
  ngsw.json
```
Service Workers

- Offline
- Update on reload
- Bypass for network
- Show all

http://127.0.0.1:8082/

Source: ngsw-worker.js
Received: 11/6/2017, 11:02:40 PM

Status: #2720 activated and is running stop

Clients: http://127.0.0.1:8082/ focus

Network:

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Status</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>GET</td>
<td>200</td>
<td>document</td>
<td>(from ServiceWorker)</td>
</tr>
<tr>
<td>styles.d41d8cd98f00b204e980.bundle</td>
<td>GET</td>
<td>200</td>
<td>stylesheet</td>
<td></td>
</tr>
<tr>
<td>inline.bcfe357835369db7a7c.bundle</td>
<td>GET</td>
<td>200</td>
<td>script</td>
<td></td>
</tr>
<tr>
<td>polyfills.ad37cd45a71ab38ee76.bundle</td>
<td>GET</td>
<td>200</td>
<td>script</td>
<td></td>
</tr>
<tr>
<td>main.40a7c44944aa4a7e5a45.bundle</td>
<td>GET</td>
<td>200</td>
<td>script</td>
<td></td>
</tr>
</tbody>
</table>
Hint #1: Checking the status

https://yourwebsite.com/ngsw/state

NGSW Debug Info:

Driver state: NORMAL ((nominal))
Latest manifest hash: cd4716ff2d3e24f4292010c929ff429d9eeead73
Last update check: 9s215u

=== Version 34c3fd2361735b1330a23c32880640feb059305 ===

Clients: 7eb10c76-d9ed-493a-be12-93f305394a77

=== Version cd4716ff2d3e24f4292010c929ff429d9eeead73 ===

Clients: ee22d69e-37f1-439d-acd3-4f1f366ec8e1

=== Idle Task Queue ===
Last update tick: 4s602u
Last update run: 9s222u
Task queue:
Adding NGSW to the existing app

1. Install the package
   
   \$ npm install @angular/service-worker --save

2. Enable build support
   
   \$ ng set apps.0.serviceWorker=true

3. Register NGSW for your app

4. Create configuration file
Service worker build support in the CLI

src/ ngsw-config.json → Build → dist/ ngsw.json

node_modules/@angular... ngsw-worker.js → Copy → dist/

Can be npm-scripted for legacy Angular CLIs!
import { ServiceWorkerModule } from '@angular/service-worker';
import { environment } from '../environments/environment';
...
@NgModule({
...
    imports: [
        ...
        ServiceWorkerModule.register('/ngsw-worker.js',
            { enabled: environment.production }
        ),
    ]
})
export class AppModule { }
NGSW configuration file

src/ngsw-config.json

```json
{
    "index": "/index.html",
    "assetGroups": [...],
    "dataGroups": [...]
}
```
App shell

assetGroups

```json
{
    "name": "app",
    "installMode": "prefetch",
    "resources": {...}
}
```
App shell resources

assetGroups / "app" / resources

"resources": {
  "files": [
    "favicon.ico",
    "index.html"
  ],
  "versionedFiles": [
    "/*.bundle.css",
    "/*.bundle.js",
    "/*.chunk.js"
  ]
}
App shell / on-demand

assetGroups

{
    "name": "assets",
    "installMode": "lazy",
    "updateMode": "prefetch",
    "resources": {...}
}
App shell / on-demand

assetGroups / "assets" / resources

"resources": {
  "files": [
    "/assets/**"
  ],
  "urls": [
    "https://fonts.googleapis.com/**",
    "https://fonts.gstatic.com/**"
  ]
}

dataGroups

{
  "name": "api-freshness",
  "urls": [
    "/api/breakingnews/**
  ],
  "cacheConfig": {
    "strategy": "freshness",
    "maxSize": 10,
    "maxAge": "12h",
    "timeout": "10s"
  }
}
dataGroups

{
    "name": "api-performance",
    "urls": [
        "/api/archive/**"
    ],
    "cacheConfig": {
        "strategy": "performance",
        "maxSize": 100,
        "maxAge": "365d"
    }
}

Runtime caching
Hint #2: Support API versioning

dataGroups

{
    "version": 2,
    "name": "api-performance",
    "urls": [
        "/api/**"
    ],
    ...
}
App version updates

Server
- v1
- v2
- v2

Browser
- v1
- v1
- v2
Hint #3: Notify about updates

import { SwUpdate } from '@angular/service-worker';

constructor(private swUpdate: SwUpdate) {}  

this.swUpdate.available.subscribe(event => {
  let snackBarRef = this.snackBar
    .open('Newer version of the app is available', 'Refresh');

  snackBarRef.onAction().subscribe(() => {
    window.location.reload();
  })
})
Working offline

Smarter networking

Staying notified

Proper app experience
import { SwPush } from '@angular/service-worker';

constructor(private swPush: SwPush) {}

subscribeToPush() {
  this.swPush.requestSubscription(
    {
      serverPublicKey: this.VAPID_PUBLIC_KEY
    }
  ).then(pushSubscription => {
    // Pass subscription object to backend
  })
}
Push notifications / send

server-side.js / send Notification payload

```
{
  "notification": {
    "title": "Very important notification",
    "body": "Angular Service Worker is cool!",
    "icon": "https://angular.io/assets/logo.png",
    "actions": [
      {
        "action": "gocheck",
        "title": "Go and check"
      }
    ]
  }
}
```
Push and notification use different, but complementary, APIs: push is invoked when a server supplies information to a service worker; a notification is the action of a service worker or web page script showing information to a user.

Recording network activity...
Perform a request or hit \( \text{R} \) to record the reload.
Note: some network activity from out-of-process iframes might be missing. See https://crbug.com/750901#c4 for more details.
Hint #4: Kill switch

1. Long way

   `ng set apps.0.serviceWorker=false`
   `ng build --prod`
   ...
   `deploy`

2. Short way

   `rm dist/ngsw.json`
   ...
   `deploy`
Hint #4: Kill switch

3. Proper way

```bash
cp dist/safety-worker.js dist/ngsw-worker.js
...deploy
```

```javascript
safety-worker.js

self.addEventListener('install', e => { self.skipWaiting(); })

self.addEventListener('activate', e => {
  e.waitUntil(self.clients.claim());
  self.registration.unregister().then(
    () => { console.log('Unregistered old service worker'); 
  });
});
```
Main available features

- App Shell
- Runtime Caching
- Push Notifications
- Smart Updates
Angular Service Worker advantages

- Essential features are config-driven
- Decoupled updates model
- Integrity checks
- Doing things in Angular way
Thank you!

Maxim Salnikov
@webmaxru
Questions?