6-PWA

Progressive Web Applications

```
# 1. Angular Service Worker con el CLI
# 2. Configuración de caché
# 3. Actualizaciones y notificaciones
```

```
As a: web user,
  I want: to have similar capabilities to native
  so that: I can  have a better user experience
```

1 Angular Service Worker con el CLI

1.1 Instalación de PWA

```
ng add @angular/pwa --project shop
```

1.2 Modificaciones automáticas

angular json

```
architect.build.options.assets: [..."apps/shop/src/manifest.json"]
architect.configuration.production.serviceWorker: true
architect.configuration.production.ngswConfigPath: "apps/shop/ngsw-config.json"
```

index.html

```
  <link rel="manifest" href="manifest.json">
  <meta name="theme-color" content="#1976d2">
  <noscript>Please enable JavaScript to continue using this application.
</noscript>
```

app.module.ts

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

otros

- ngsw-config.json
- manifest.webmanifest
- /assets/icons

1.3 Paquetes recomendados

```
yarn add -D ngx-pwa-icons
cd apps/shop
apps/shop/icon.png
ngx-pwa-icons
"build:shop": "ng build shop --prod",
yarn add -D angular-http-server
"start:shop:pwa": "angular-http-server --open -p 8080 --path ./dist/apps/shop",
```

1 Angular Service Worker con el CLI

Instalación

Modificaciones automáticas

Paquetes recomendados

---

class: impact

2 Configuración de caché

2.1 Assets

default Full App

```
  "assetGroups": [
    {
      "name": "app",
      "installMode": "prefetch",
      "resources": {
        "files": [
          "/favicon.ico", "/index.html",
          "/manifest.webmanifest",  "/*.css", "/*.js"
        ]
      }
    },
    {
      "name": "assets",
      "installMode": "lazy",
      "updateMode": "prefetch",
      "resources": {
        "files": [
          "/assets/**",
          "/*.(eot|svg|cur|jpg|png|webp|gif|otf|ttf|woff|woff2|ani)"
        ]
      }
    }
  ],
```

---

proposed Lazy App

```
    "assetGroups":[
      {
        "name": "coreapp",
        "installMode": "prefetch",
        "resources": {
          "files": [
            "/favicon.ico",
            "/index.html",
            "/manifest.webmanifest",
            "/*.css",
            "/common*.js",
            "/main*.js",
            "/ngsw*.js",
            "/*woker*.js",
            "/*polyfills*.js",
            "/runtime*.js"
          ]
        }
      },
      {
        "name": "lazyapp",
        "installMode": "lazy",
        "updateMode": "prefetch",
        "resources": {
          "files": ["/*.js"]
        }
      }...
```

2.2 API

dataGroups

```
 "dataGroups": [
  {
    "name": "cache-first-greeting",
    "urls": [
      "http://localhost:3333/api"
    ],
    "cacheConfig": {
      "strategy": "performance",
      "maxAge": "1d",
      "maxSize": 10
    }
  },
  {
    "name": "api-first-rates",
    "urls": [
      "https://api.exchangeratesapi.io/latest?symbols=GBP",
      "https://api.exchangeratesapi.io/latest"
    ],
    "cacheConfig": {
      "strategy": "freshness",
      "timeout": "5s",
      "maxAge": "1h",
      "maxSize": 10,
    }
  }
]
```

3 Actualizaciones y notificaciones

3.1 Actualización de versiones

`apps\shop\ngsw-config.json`

```
"appData": {
  "version": "6.0.1",
  "changelog": "PWA: trying a better update mode"
},
```

`apps\shop\src\app\app.component.ts`

```
constructor(private swUpdate: SwUpdate) {
  this.checkVersionUpdates();
}
```

```
private checkVersionUpdates() {
  if (this.swUpdate.isEnabled) {
    this.swUpdate.available.subscribe(event => {
      if (event.current.appData) {
        const appData: any = event.current.appData;
        let msg = `New version ${appData.version} available.`;
        msg += `${appData.changelog}.`;
        msg += 'Reaload now?';
        if (confirm(msg)) {
          window.location.reload();
        }
      }
    });
  }
}
```

`apps\shop\ngsw-config.json`

```
"appData": {
  "version": "6.1.0",
  "changelog": "PWA: Added a better update mode"
},
```

3.2 Notificaciones Push

```
constructor(private swPush: SwPush) {
  this.subscribeToNotifications();
}
  private subscribeToNotifications() {
    if (this.swPush.isEnabled) {
      this.swPush
        .requestSubscription({ serverPublicKey: 'VAPID_PUBLIC_KEY' })
        .then(sub => {
          console.log('subscription to server', sub.toJSON());
          this.swPush.messages.subscribe(msg => console.log('Received: ', msg));
        })
        .catch(err => console.error('Could not subscribe', err));
    }
  }
```

> **Blog de apoyo:** PWA, Entre la web y las apps con Angular
>
> > By Alberto Basalo

> Next:

Server Side Rendering

Angular Universal

Despliegue con Node Express

Variantes: shell y pre-rendering