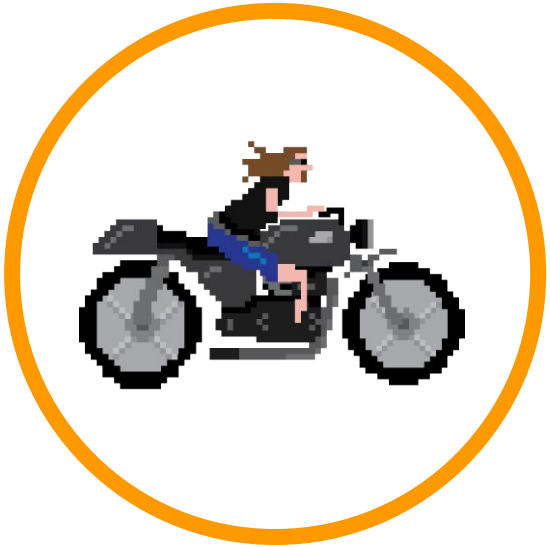
PWA & Angular

...e la tua SPA diventa PWA
in pochi secondi!

Francesco Sciuti @ FullStackConf 2019

https://www.academy.it/intro
https://www.francescosciuti.it

acadevmy



Chi sono?

Developer per scelta e per passione,
amante di nerdaggini di ogni tipo
ed amante della condivisione del sapere!

Da non troppo anche (quasi)
imprenditore!

academvmy

Software & Learning Factory

Web | Mobile | DevOps | Labs | Formazione

Training | Eventi | Talks

academy

ma basta presentazioni e parliamo di...

academy

Progressive Web App

Francesco Sciuti @ FullStackConf 2019

https://www.academy.it/intro
https://www.francescosciuti.it

academy

Cosa sono le PWA?

“Progressive Web App è un termine che si riferisce ad applicazioni web che vengono sviluppate e caricate come normali pagine web, ma che si comportano in modo simile alle applicazioni native quando utilizzate su un dispositivo mobile”

Wikipedia

academy

Cosa sono le PWA?

“Progressive Web App sono delle comuni web application che sfruttano delle tecnologie (alcune neanche così nuove) che consentono all’utente di vivere un’esperienza molto simile a quella nativa, che sia essa mobile o desktop”

Me medesimo

Ciccio Sciuti 1 - 0 Wikipedia
Tiè!

Cosa distingue una PWA?

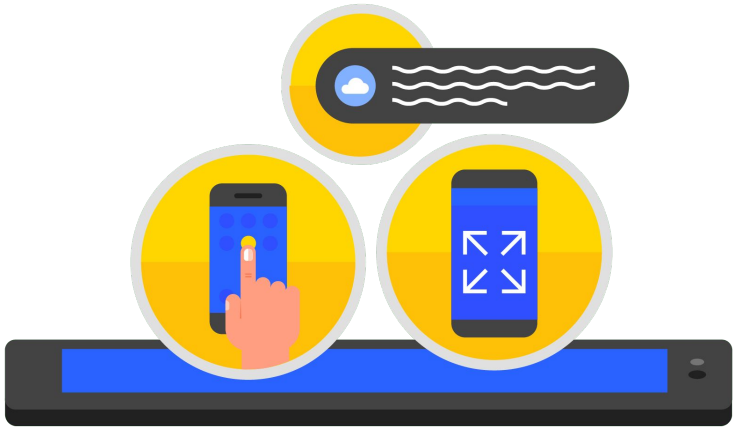
- Affidabilità
  Funzionano anche in caso di instabilità di rete
- Rapidità
  Si caricano istantaneamente e rispondono rapidamente alle interazioni dell'utente
- Coinvolgimento
  Sembrano applicazioni native sul dispositivo e offrono un'esperienza utente ottimale

academvmy

Quali caratteristiche hanno?

- **Progressive**
  Grazie al Progressive Enhancement da cui ereditano parte del nome
- **Responsive**
  Si adattano quindi ai dispositivi che le ospitano
- **Network Independent**
  Possono lavorare offline!
- **App-like**
  Possono essere esteticamente uguali alle app native, quindi ad esempio a tutto schermo
- **Continuous Update**
  Hanno un processo di aggiornamento molto più semplice in quanto web application
- **Safe (HTTPS)**
  Viaggiano su protocollo sicuro
- **Discoverable**
  Facilmente identificabili come applicazioni ma rintracciabili dai motori di ricerca
- **Re-engageable**
  Possono richiamare l'interesse degli utenti tramite le Push Notification
- **Installable**
  Possiamo installarle sui nostri dispositivi come app native
- **Linkable**
  Ci basta un url per raggiungerle

Come fanno a fare tutto ciò?



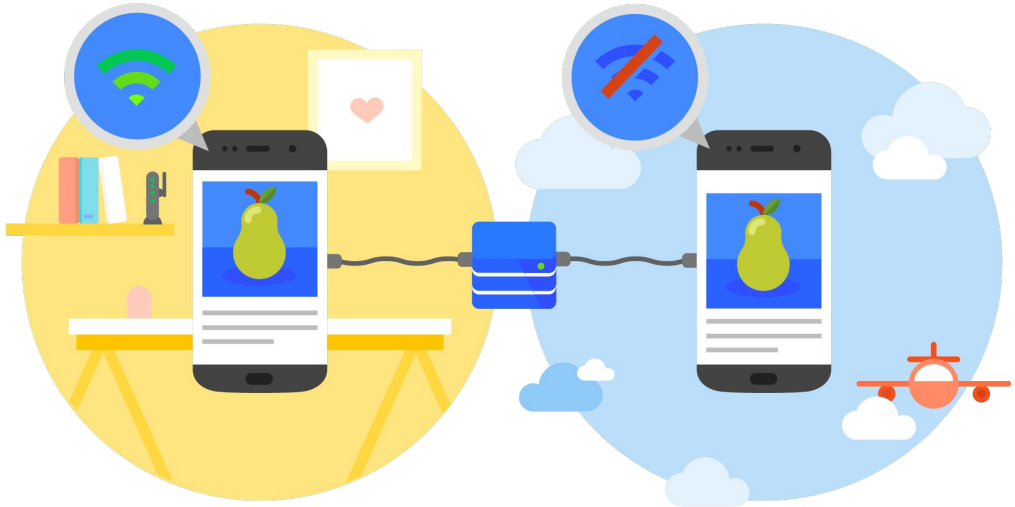
- Add to Home Screen
  (o A2HS perché gli acronimi fanno sempre figo!)
- Cache Management
- Push Notification
- App Shell
- Devices Interaction
- Persistance
- ...

academvmy

Con quali tecnologie?

- Web Manifest
- Service Workers
- Server Side Rendering
- Web API Push API, Notification API, Cache API, Intersection Observer, etc...
- Client-side Storage IndexedDB
- Native App Integration Trusted Web Activities/WebAPK/APPX
- HTTPS

academvmy



Ripassiamo rapidamente le principali...

academy

Manifest

Un semplice file JSON che consente di descrivere il comportamento della nostra PWA

Il file manifest contiene tutte le informazioni necessarie all'ambiente che dovrà eseguire la nostra app, consentendo quindi all'environment ospitante di sapere quali sono il titolo o le icone da utilizzare per installare l'applicazione, lo scope di esecuzione, alcuni aspetti estetici, etc...

academy

Manifest

Comunemente il file ha estensione *.webmanifest*

- Nome
- Icone
- Url di avvio
- Colore di fondo e del tema
- Modalità di visualizzazione
- Scope di esecuzione
- Orientamento
- Splash Screen

[MDN Manifest]

```
{
  "short_name": "Maps",
  "name": "Google Maps",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/maps/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/maps/",
  "theme_color": "#3367D6"
}
```

academy

Service Workers

Il cuore delle nostre PWA

Sposando alcune API consentono di gestire funzionalità quali le notifiche push, la gestione offline e la sincronizzazione in background.

Francesco Sciuti @ FullStackConf 2019

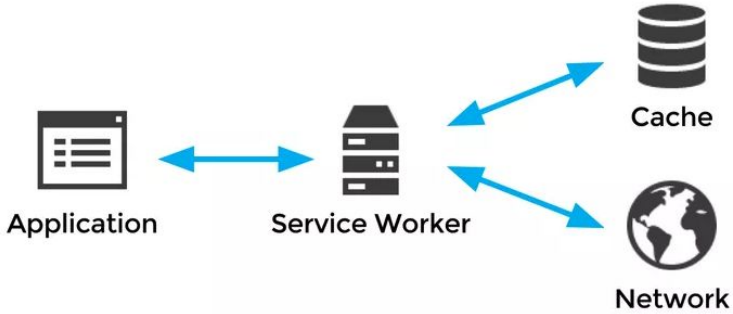
https://www.academy.it/intro
https://www.francescosciuti.it

academvmy

Service Workers

Sono degli script che vengono eseguiti dal browser in processi separati rispetto alla pagina che li registra e che lavorano quindi in background.

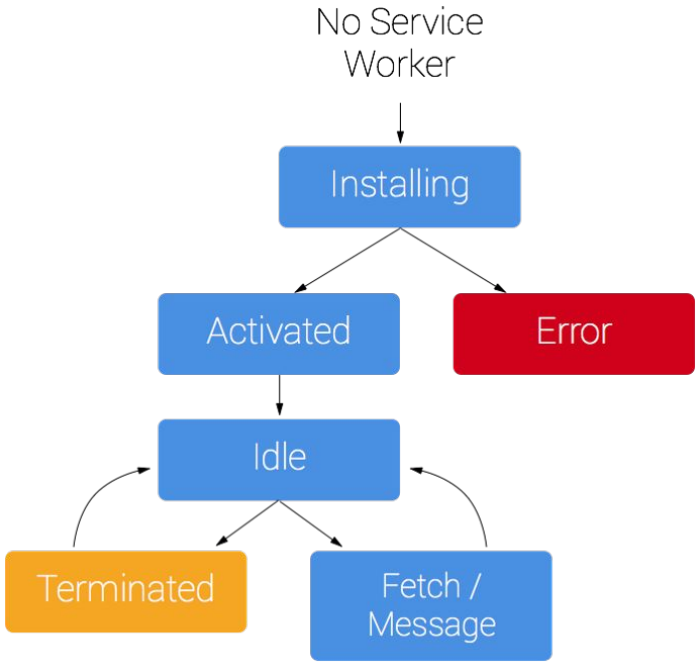
I SW lavorano come dei proxy ed intercettano tutte le richieste HTTP dell'applicazione e decidono come rispondere secondo la logica che svilupperemo.



academvmy

Service Workers

- Gestione della Cache
- Gestione Push Notification
- Sync dei dati in background
- Lifecycle programmabile
- Processo separato e con un proprio scope di esecuzione



MDN - Service Workers Overview
MDN - Service Workers Offline Implementation

academvmy

App Shell

Il minimo indispensabile per un rendering veloce

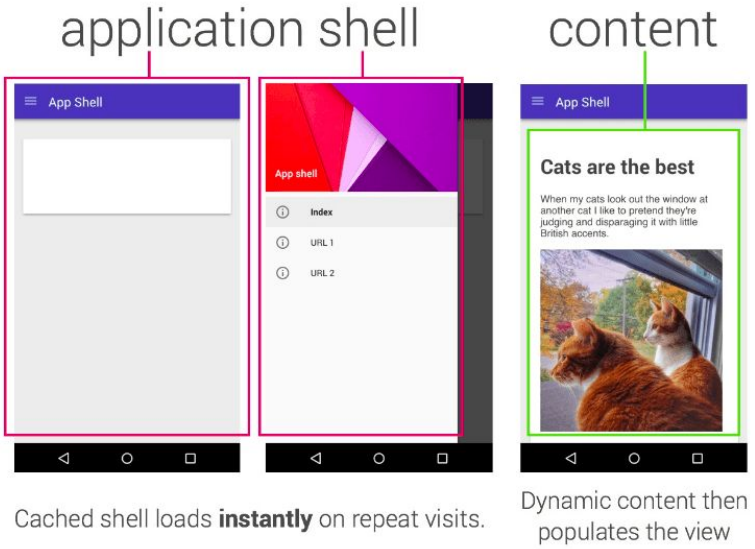
Per App Shell intendiamo tutto l'HTML, CSS e JS minimo ed indispensabile utile per essere renderizzato immediatamente dal browser in modo da offrire all'utente un immediato feedback visivo ed evitare l'abbandono della nostra PWA.

academvmy

App Shell

I Service Worker (sempre loro!) memorizzano l'interfaccia di base (la shell) della web application in modalità Responsive Web Design.

Questa shell fornisce un layout statico iniziale,nella quale il contenuto può essere caricato sia progressivamente che dinamicamente immediatamente.



https://www.academvmy.it/intro
https://www.francescosciuti.it

academvmy

Push Notification

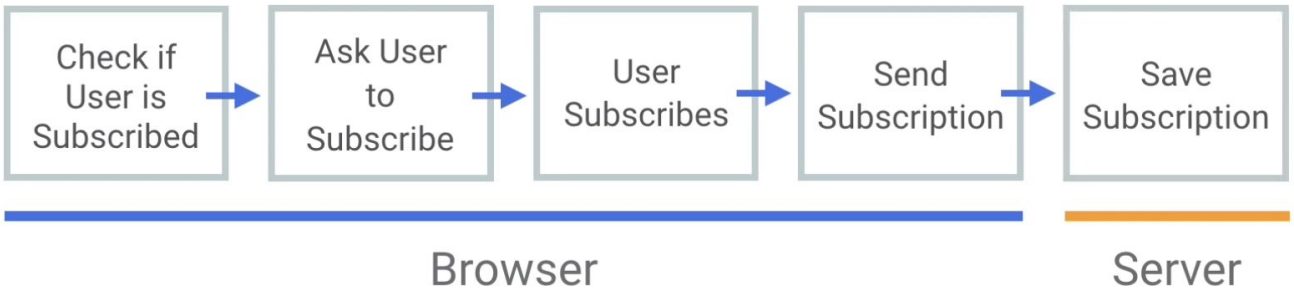
Sebbene non siano ancora supportati su iOS, le nostre PWA possono inviare notifiche push al browser con l'API Push.

L'API Push viene utilizzata insieme alle Notification API per visualizzare il messaggio.

Il vantaggio è che l'API Push consente la consegna della notifica anche quando l'utente non naviga sulla PWA, poiché sono *build on top* sui Service Workers.

academvmy

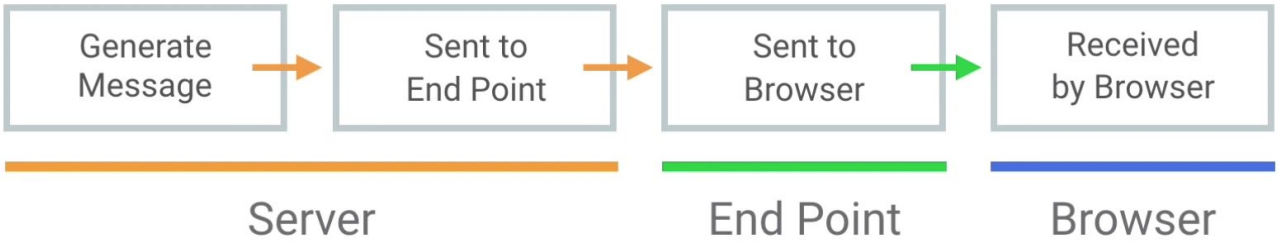
Push Notification



Francesco Sciuti
https://www.academvmy.it/intro

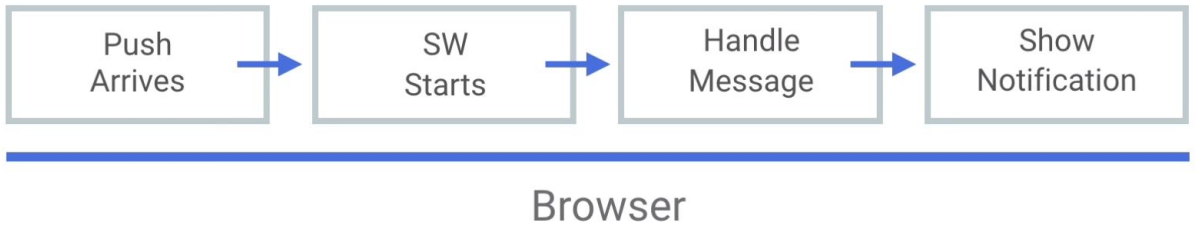
academvmy

Push Notification



academvmy

Push Notification



Francesco Sciuti
https://www.academvmy.it/intro

academy

Web API

Gli ingredienti indispensabili per un'ottima ricetta

Le Web API sono ormai molteplici e supportate rapidamente da tutti i browser sul mercato, e molte di esse sono fondamentali per creare PWA e far si che siano appunto *progressive*.

Francesco Sciuti
https://www.academy.it/intro

Qualche API da usare

- ## Service Worker API
  Utili per la gestione *proxy* dello stato del network e della comunicazione server.
- ## Notification/Push API
  Due API usate per configurare e mostrare le notifiche all'utente, ed iscriversi a un servizio push e ricevere messaggi push
- ## Cache API
  Possiamo installarle sui nostri dispositivi come app native
- ## Fetch API
  Un modo flessibile e potente per ottenere risorse locali o remote.
- ## Intersection Observer
  Consente di configurare un callback che viene triggerato ogni volta che un elemento, chiamato target, interseca il viewport o un ulteriore elemento specificato.
- ## FormData API
  Fornisce un modo per costruire facilmente un insieme di coppie chiave / valore che rappresentano i campi modulo e i loro valori, da utilizzare con il metodo XMLHttpRequest.send ()
- ## Channel Messaging API
  Consente l'esecuzione di due script separati in contesti di navigazione diversi allegati allo stesso documento.
- ## Background Sync
  Consente di rimandare le azioni fino a quando l'utente non ha una connettività stabile.
- ## Payment Request API
  Standard aperto che sostituisce i flussi di pagamento tradizionali consentendo ai commercianti di richiedere e accettare qualsiasi pagamento in una singola chiamata API.
- ## Streams API

academy

Angular

Francesco Sciuti @ FullStackConf 2019

https://www.academy.it/intro
https://www.francescosciuti.it

academy

Angular

One framework. Mobile & desktop.

- Cross Platform
  (PWA, Native, Desktop)
- Veloce e Performante
  (Code Generation, Universal, Code Splitting)
- Produttività
  (Templates, Angular CLI, IDEs)
- Pensato per la Full Dev Story
  (Testing, Animation, Accessibility)

Angular

Dalla versione 5 del Framework è stato aggiunto il supporto ai Service Workers, che hanno aperto quindi le porte alla creazione di PWA su Angular in maniera semplice e rapida.

Il logo figo invece è arrivato dopo... :D

academy

Angular

Come creare una PWA

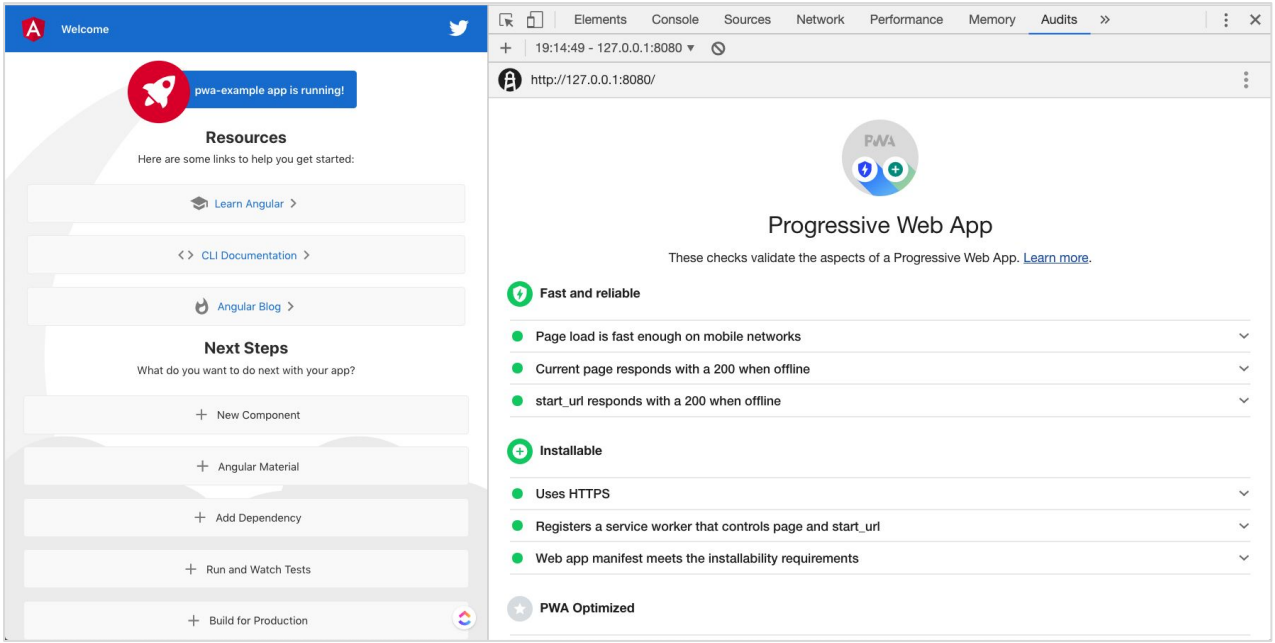
- `ng new pwa-test`
- `ng add @angular/pwa [--project pwa-test]`
  Durante questo processo vengono creati e modificati alcuni files di progetto come index.html, package.json, angular.json, ngsw-config.json, etc...
  (il tutto grazie a Schematics).
- `ng build --prod`
- `http-server -p 8080 -c-1 dist/pwa-test`
  N.B. Il pacchetto *http-server* va installato separatamente con il comando: *npm install -g http-server*.
- Navigare col browser su http://localhost:8080/
  N.B. È consigliata la *incognito mode*

academy

Angular

La PWA è stata creata ed ha già tutti i requisiti richiesti!

Per verificarlo usiamo *Lighthouse*
ad esempio dai Chrome Dev Tools



Angular

Come configurare una PWA

La configurazione dell'applicazione PWA è presente nel file di progetto ***manifest.webmanifest***

Il file è un file manifest standard a tutti gli effetti e quindi ha tutte le chiavi utili per le PWA

La configurazione per la gestione di tutti i comportamenti dei SW è presente nel file di progetto ***ngsw-config.json***

Il file in fase di build viene poi lavorato dalla CLI per produrre i comportamenti necessari del SW.

Analizziamo il progetto!

(in caso di allarme usare il *Piano B*)

academvmy

Angular

Come configurare una PWA (ngsw-config.json)

La chiave ***assetGroups*** è utilizzata per le richieste HTTP vengono memorizzate nella cache dal SW.
È utilizzata per:

- files di progetto (html, css, js)
- gli assets (come ad esempio le immagini)

Ogni gruppo può avere strategie di caching differenti (prefetch | lazy).

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

academvmy

Angular

Come configurare una PWA (ngsw-config.json)

La chiave ***dataGroups*** è utilizzata per le richieste HTTP che non sono sottoposte a versione insieme all'app.

È utilizzata per:

- Chiamate API
- Risorse non versionate

Ogni gruppo può avere strategie di caching differenti (freshness | performance).

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

Angular

Cosa viene prodotto dal processo di build

ngsw-worker.js

È il vero e proprio SW ed è istanziato (su un processo separato) da Angular per mezzo di *navigation.serviceWorker.register()*

N.B. Il file ngsw-worker.js sarà sempre lo stesso per ogni build e rimarrà quindi lo stesso fino all'aggiornamento a una nuova versione di Angular.

ngsw.json

Questo è il file di configurazione che verrà utilizzato dal SW.
Questo file è basato sulle configurazioni indicate sul file *ngsw-config.json*

La versione dell'app è determinata dal contenuto del file ngsw.json, che include hash per tutto il contenuto monitorato. Se anche uno dei files memorizzati nella cache cambia, l'hash del file cambierà in ngsw.json, facendo sì che il SW si accorga di una nuova versione disponibile.

Angular

Il *ServiceWorkerModule*

Il ***ServiceWorkerModule*** è il modulo che gestisce e che ci consente di dialogare con il SW ed offre due interessantissimi servizi:

- ***SwUpdate*** per la gestione degli updates delle versioni della nostra applicazione
- ***SwPush*** per la gestione delle Push Notification

Angular

Creare una App Shell

Angular consente la generazione di una App Shell per mezzo della CLI ed utilizza *Universal* per la renderizzazione del contenuto iniziale.

- `ng generate app-shell --client-project my-app --universal-project server-app`
- `ng run my-app:app-shell --configuration=production`

Angular

Come funziona il caching del Service Worker

- Il Caching dei files dell'applicazione viene valutato come una singola unità, e difatti **i files sono aggiornati tutti insieme**.
  Un'applicazione in esecuzione continua a essere eseguita con la stessa versione di tutti i file. Non inizia improvvisamente a ricevere file memorizzati nella cache da una versione più recente, dato che potrebbero essere potenzialmente incompatibili.

- Quando gli utenti aggiornano l'applicazione, vedono **l'ultima versione completamente memorizzata nella cache.**
  Gli aggiornamenti avvengono in background, relativamente rapidamente dopo la pubblicazione delle modifiche. La versione precedente dell'applicazione viene pubblicata fino a quando un aggiornamento è pronto dopo essere stato installato.

academvmy

Angular

Come funziona il caching del Service Worker

- Ogni volta che viene distribuita una nuova build dell'app, **anche se viene aggiornato solo un singolo file, il SW tratta tale build come una nuova versione dell'app**.

- **Il SW fornisce una garanzia: un'app in esecuzione continuerà a eseguire la stessa versione dell'app**. Se viene aperta un'altra istanza dell'app in una nuova tab del browser Web, verrà invece offerta la versione più recente dell'app.
  Di conseguenza, quella nuova scheda può eseguire una versione diversa dell'app rispetto alla scheda originale.

academy

Angular

Come funziona il caching del Service Worker

- **Il SW viene scaricato quando l'app viene aperta per la prima volta** e quando si accede all'app dopo un periodo di inattività.
  Se il SW è stato modificato, verrà aggiornato in background.

- È possibile **ignorare il SW per alcune richieste** ed è sufficiente impostare *ngsw-bypass* come *header* della richiesta o come parametro di query.

Angular

Come vengono installare le versioni dell'app

- Ogni volta che l'utente ricarica l'applicazione, **il SW verificherà se sul server è disponibile un nuovo file *ngsw.json***.
  La precedente e la nuova versione del file ngsw.json verranno confrontate e il nuovo bundle verrà scaricato e installato in background.
  È pertanto essenziale verificare frequentemente se sul server è disponibile una nuova versione dell'applicazione.

- Se viene trovato un aggiornamento, questo viene scaricato e memorizzato nella cache automaticamente e **viene pubblicato al successivo caricamento dell'applicazione.**

academy

Angular

Debug e Disattivazione del Service Worker

Debug

Il SW espone sotto un path virtuale le informazioni di debug
ed attualmente il path è *ngsw/state*.

Disattivazione del SW

È possibile farlo seguendo due strategie distinte:
Fail-safe o *Safety Worker*

academy

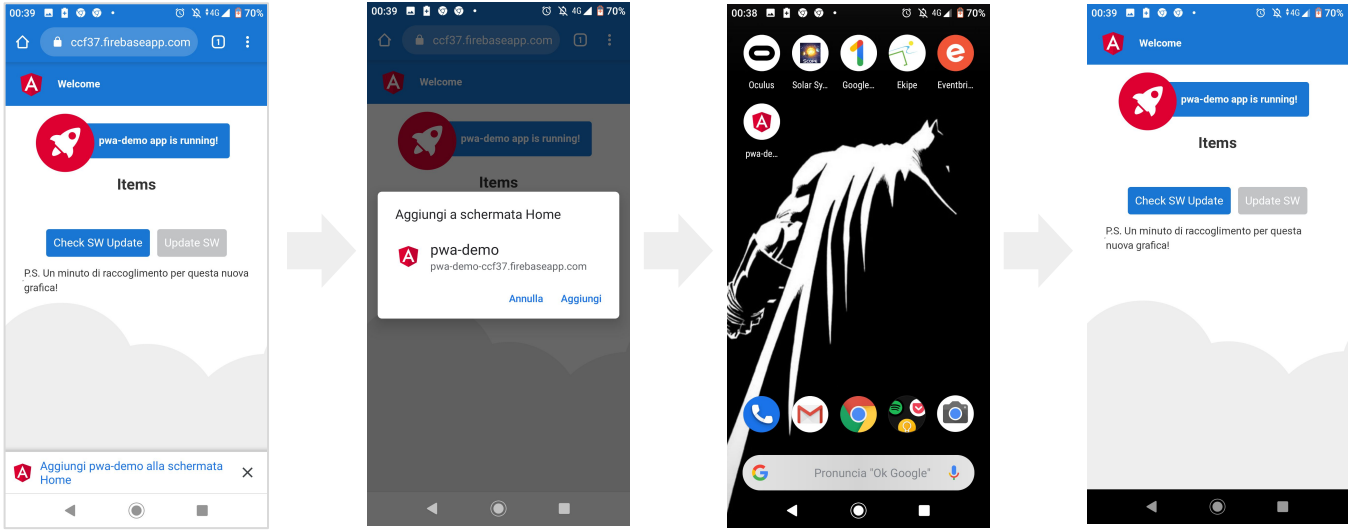
Mobile e Desktop

Le PWA offrono un'esperienza *native-like* sia su mobile che su desktop.

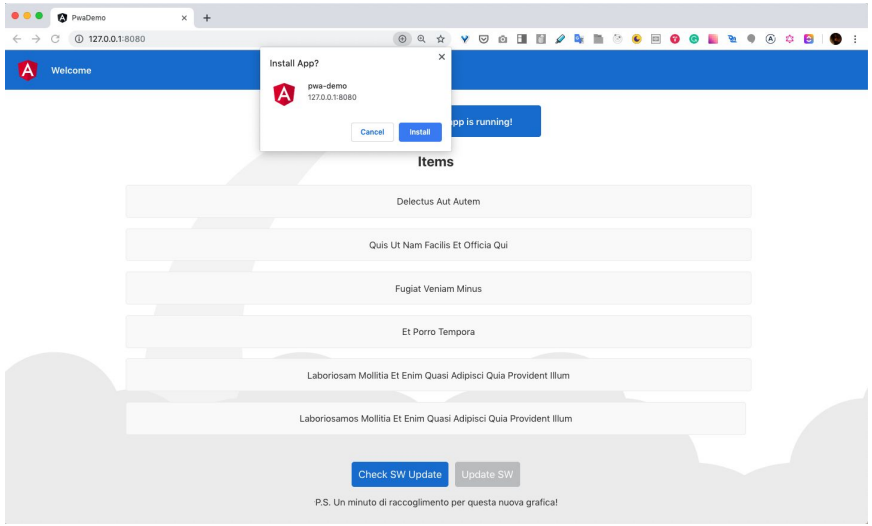
Attualmente il supporto è offerto da Chrome...
ma ci si sta lavorando anche da altri fronti!

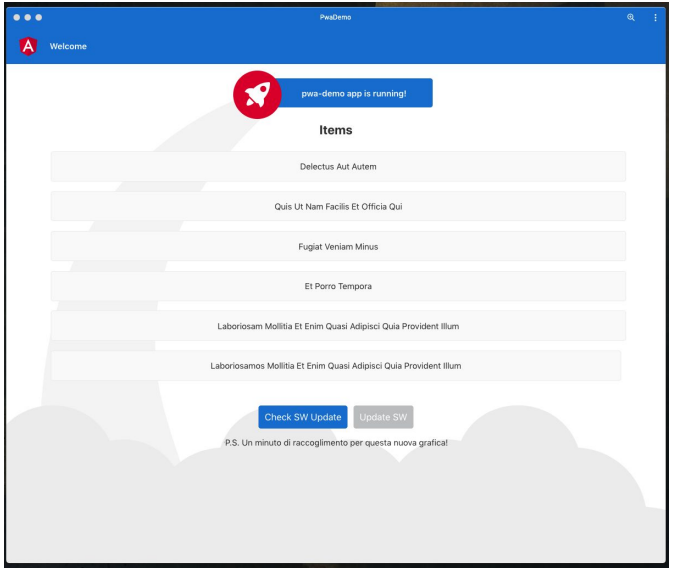
academy

Mobile e Desktop



Mobile e Desktop





academy

Gli store

La pubblicazione sugli store di applicazioni sono ufficialmente supportate su *Google Play Store* e *Windows Store*

In queste occasioni sarà necessario lavorare su tecnologie come TWA (Trusted Web Activity) o APPX

Per Apple è necessario invece continuare a lavorare come avremmo fatto con un'app Cordova... con tutte le limitazioni che impone l'App Store.

academvmy

La compatibilità

I browser sono molto avanti nell'implementazione delle tecnologie utili per le PWA ma ancora la compatibilità non è completa (soprattutto su IOS).

- Add to Home Screen
- Splash Screen
- Supporto alle Push Notification
- Ciclo di Vita della PWA
- Orientation
- Supporto parziale alle gesture
- Supporto parziale al Manifest
- Web API disponibili
- Tags specifici per IOS

academvmy

I tools

PWA
COMPAT

Qualche Links

Docs

- Google Developer PWA Docs
- MDN PWA Docs
- ServiceWorkers Cookbook
- PWA Checklist
- IsServiceWorkerReady?
- PWA & IOS 12.2
- PWA & IOS 11.3

Tools

- Lighthouse Report Viewer
- PWABuilder | PWA Starter Kit
- Workbox | Workbox in NG
- Web-Push

Case Studies

- Pinterest PWA
- Maps Go
- Tinder PWA
- Twitter Lite

Examples / Labs

- Your First PWA
- Google PWA Training
- HNPWA

Directory

- PWA Stats
- AppScope

Grazie a tutti!

Se avete voglia di dare un'opinione, di contattarmi o di vincere un premio...
(scherzo...non c'è alcun premio ma faceva molto figo! :D)

https://www.academvmy.it/intro
https://www.francescosciuti.it