



PWA

Progressive Web App





“

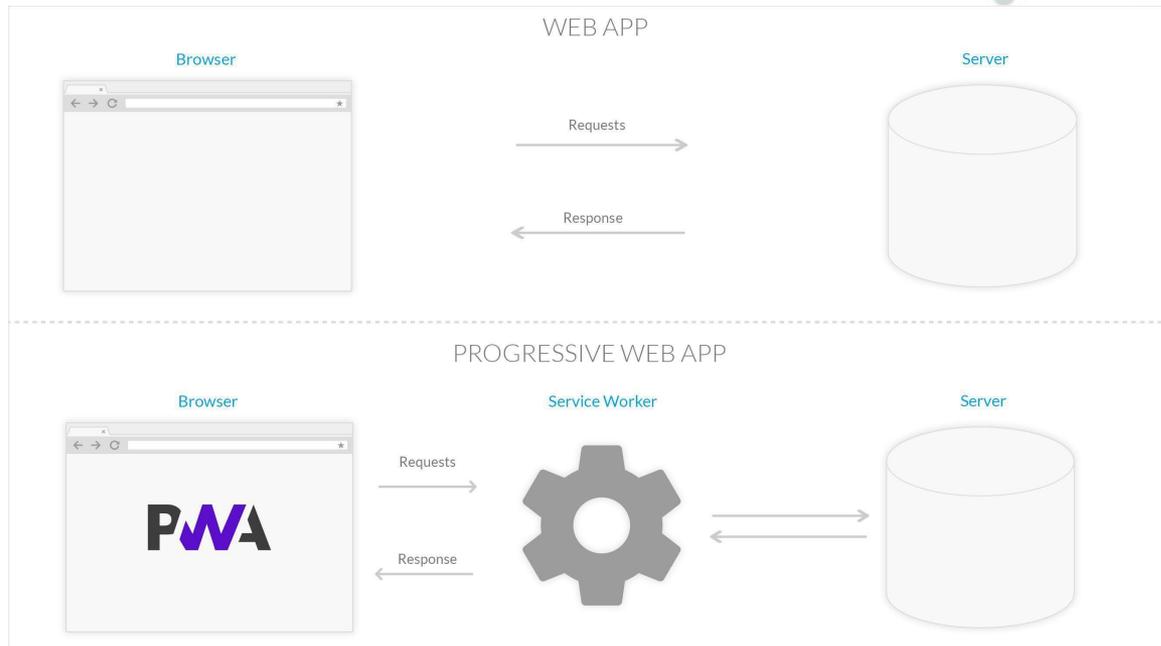
Il termine **Progressive Web App** (PWA, applicazioni web progressive) viene utilizzato per indicare una nuova metodologia per sviluppare software. Diversamente dalle applicazioni tradizionali, le progressive web apps sono un ibrido tra le normali pagine web (o siti web) e le applicazioni mobili. Questo nuovo modello di applicazioni cerca di combinare le possibilità offerte dalla maggior parte dei moderni browser con i benefici dell'utilizzo in mobilità.

https://it.wikipedia.org/wiki/Progressive_web_app

Few Requirements for PWA

- HTTPS
- Service Worker
- App Shell
- App manifest
- Connectivity-independent

PWA: Service Worker



- Intercepting network requests
- Caching
- retrieving resources from the cache
- delivering push messages

<https://developers.google.com/web/tools/workbox>

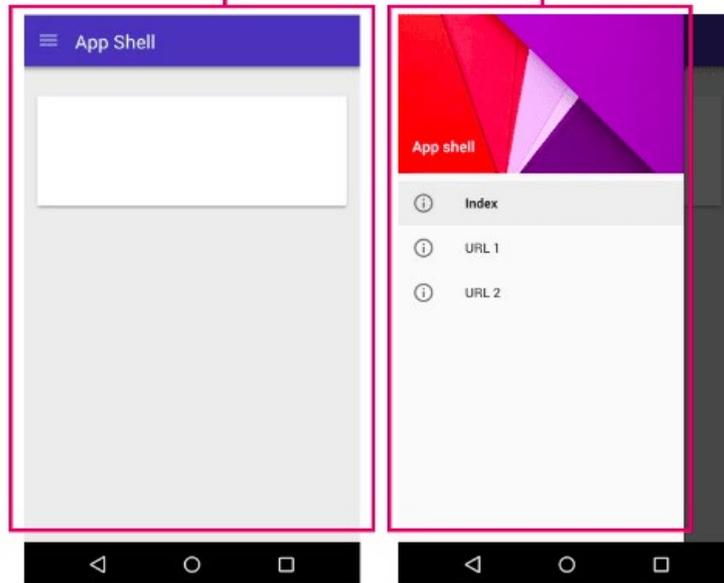
<https://ionicframework.com/pwa>

https://developers.google.com/web/fundamentals/codelabs/offline#top_of_page

https://blog.goodbarber.com/it/I-Service-Worker_a555.html

PWA: App shell

application shell



Cached shell loads **instantly** on repeat visits.

content



Dynamic content then populates the view

<https://developers.google.com/web/fundamentals/architecture/app-shell?hl=it>

PWA: Manifest

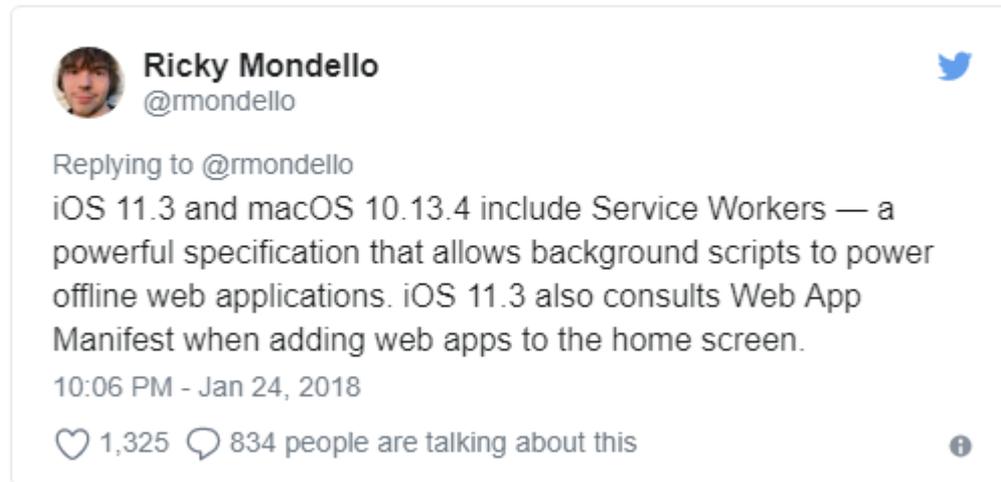
```
{
  "short_name": "AirHorner",
  "name": "Kinlan's AirHorner of Infamy",
  "icons": [
    {
      "src": "launcher-icon-1x.png",
      "type": "image/png",
      "sizes": "48x48"
    },
    {
      "src": "launcher-icon-2x.png",
      "type": "image/png",
      "sizes": "96x96"
    },
    {
      "src": "launcher-icon-4x.png",
      "type": "image/png",
      "sizes": "192x192"
    }
  ],
  "start_url": "index.html?launcher=true"
}
```



<https://developers.google.com/web/fundamentals/web-app-manifest/>
<https://developer.mozilla.org/en-US/docs/Web/Manifest>

PWA: Safari

Update : ios 11.3 supports web app manifest and service worker



A screenshot of a Twitter post by Ricky Mondello (@rmondello). The post is a reply to another user with the same handle. The text of the tweet discusses the support for Service Workers and Web App Manifest in iOS 11.3 and macOS 10.13.4. The tweet includes a timestamp of 10:06 PM on Jan 24, 2018, and shows 1,325 likes and 834 replies. The Twitter logo is visible in the top right corner of the tweet card.

Ricky Mondello
@rmondello

Replying to @rmondello

iOS 11.3 and macOS 10.13.4 include Service Workers — a powerful specification that allows background scripts to power offline web applications. iOS 11.3 also consults Web App Manifest when adding web apps to the home screen.

10:06 PM - Jan 24, 2018

♥ 1,325 💬 834 people are talking about this

Tweet about Service worker and manifest support

<https://medium.com/awebdeveloper/progressive-web-apps-pwas-are-coming-to-a-safari-near-you-216812aba5a>

PWA: Sample

Table of Known Patterns for Building PWAs

Use-case	Patterns	Examples
Publishing	Full SSR	https://babe.news/ https://ampbyexample.com https://ampproject.org
Publishing	Application Shell	https://app.jalantikus.com/ https://m.geo.tv/ https://app.kompas.com/ https://www.nfl.com/now/ https://www.chromestatus.com
Publishing	AppShell + SSR content for entry pages	https://react-hn.appspot.com https://www.polymer-project.org/1.0/
Publishing	Streams for body content / UI	https://wiki-offline.jakearchibald.com/wiki/The_Raccoons
Social	AppShell	https://web.telegram.org/
E-commerce	Application Shell	https://m.aliexpress.com/ https://kongax.konga.com/ https://m.flipkart.com (mobile/emulate) https://m.airberlin.com/en/pwa https://shop.polymer-project.org/
E-commerce	AppShell + SSR content for entry page	https://selio.com/ (try on mobile/emulate) https://lite.5milesapp.com/ (partial)
Conference	AppShell	https://events.google.com/io2016/schedule

Top PWA examples

Table of Contents

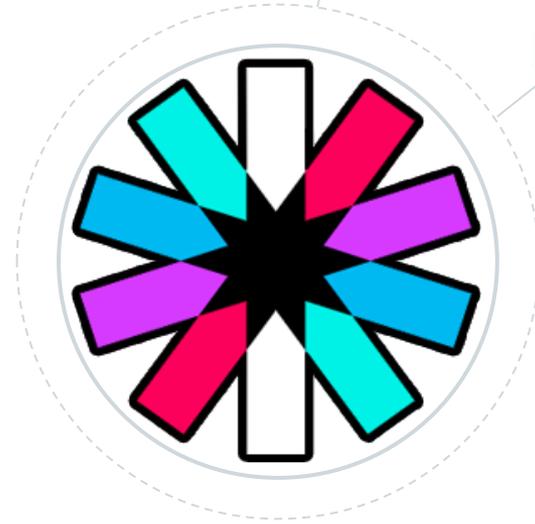
- I. Trivago Hotel Booking
- II. Pinterest
- III. Tinder
- IV. 9Gag
- V. OLX
- VI. Starbucks
- VII. Forbes

<https://appmaker.xyz/pwa-examples-successful-progressive-web-apps/>

https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures#table_of_known_patterns_for_building_pwas

JWT

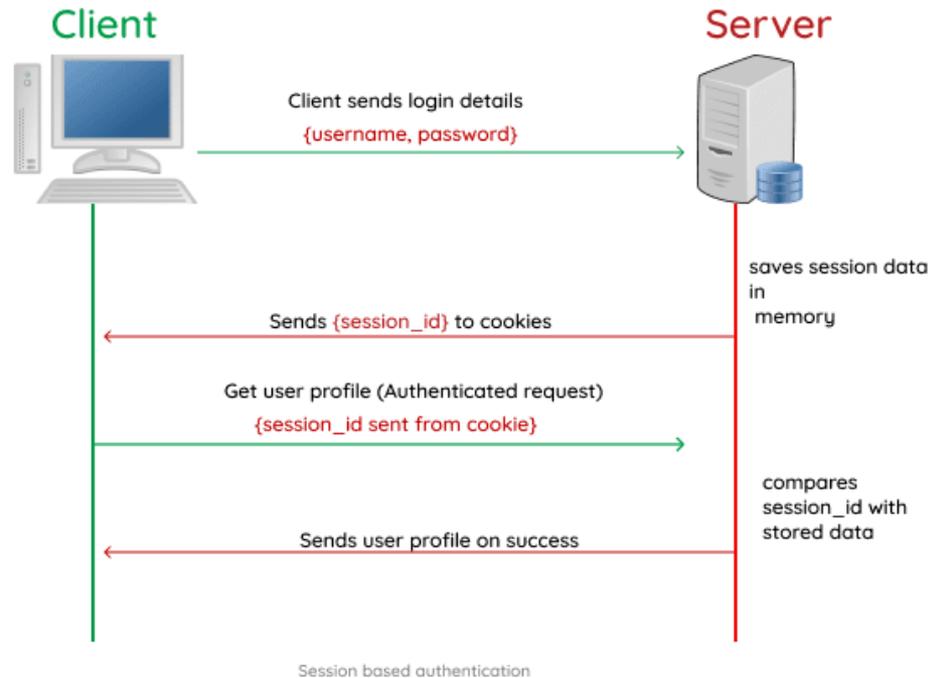
JWTs can be used as OAuth 2.0 [Bearer Tokens](#)



JSON Web Token (JWT) rfc7523

Perché è stato introdotto?

Utilizzo classico delle sessioni



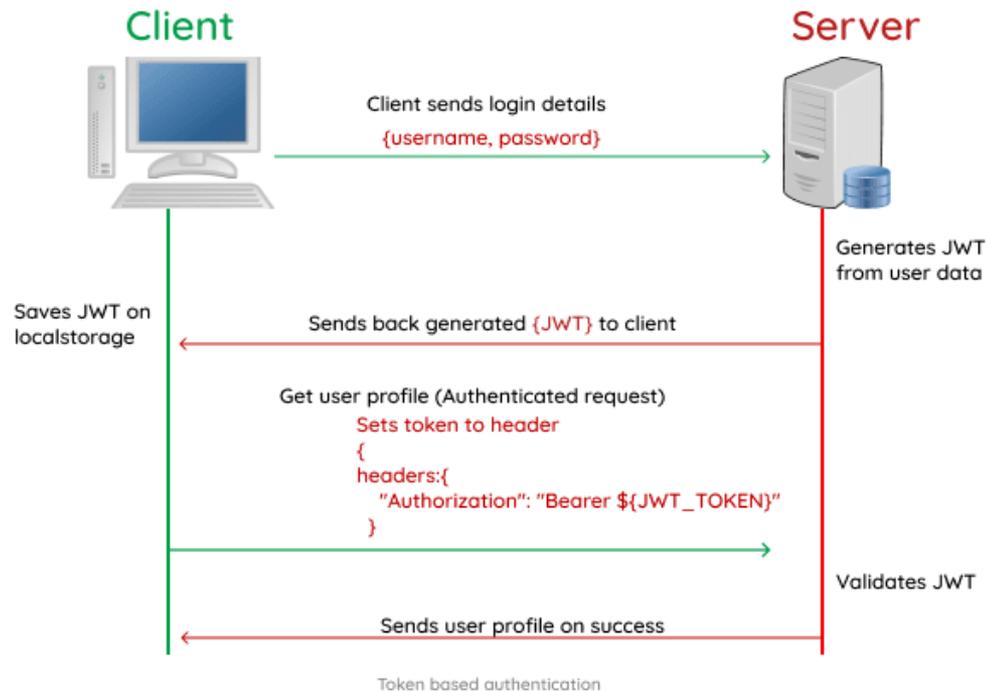
Svantaggio delle sessioni:

- Devo andare sul database a verificare la validità della sessione utente
- Nelle architetture a microservizi l'uso della sessione non è agevole
- Non scalabile con architetture a cluster
- Su una SPA ho un maggiore tempo di sviluppo rispetto ad architettura MPA

JSON Web Token (JWT) rfc7523

Perché è stato introdotto?

Utilizzo del token



Differenze?

- Non ho storage di sessioni
- Non devo gestire sessioni «morte» lato server
- I servizi possono essere realmente statefull

Ok ma dove è la vera differenza?

JSON Web Token (JWT) rfc7523

<https://jwt.io/>

Cosa posso mettere nel JWT?

- Informazioni riguardante l'utente
- Informazioni sui permessi dell'utente
- Una data di scadenza
- Una signature per validare il JWT
- Qualsiasi altra informazione

NIENTE PASSWORD NEL JWT!

JWT
JSON WEB TOKEN



HEADER
ALGORITHM
& TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

+

PAYLOAD
DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

+

SIGNATURE
VERIFICATION

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),secretKey)
```

NORDICAPIS.COM

Lo scopo di un JWT non è di crittografare i dati, quindi evitare la lettura di dati sensibili durante il trasporto (esiste SSL), ma consente alla parte ricevente di fidarsi che i dati ricevuti sono rimasti inalterati durante il trasporto.