

# Security

resources



<https://www.shodan.io/>

<https://www.exploit-db.com/>

<https://worldofvnc.net/>

<https://haveibeenpwned.com/>

<https://www.troyhunt.com/data-enrichment-people-data-labs-and-another-622m-email-addresses/>





**PWA**

Progressive Web App





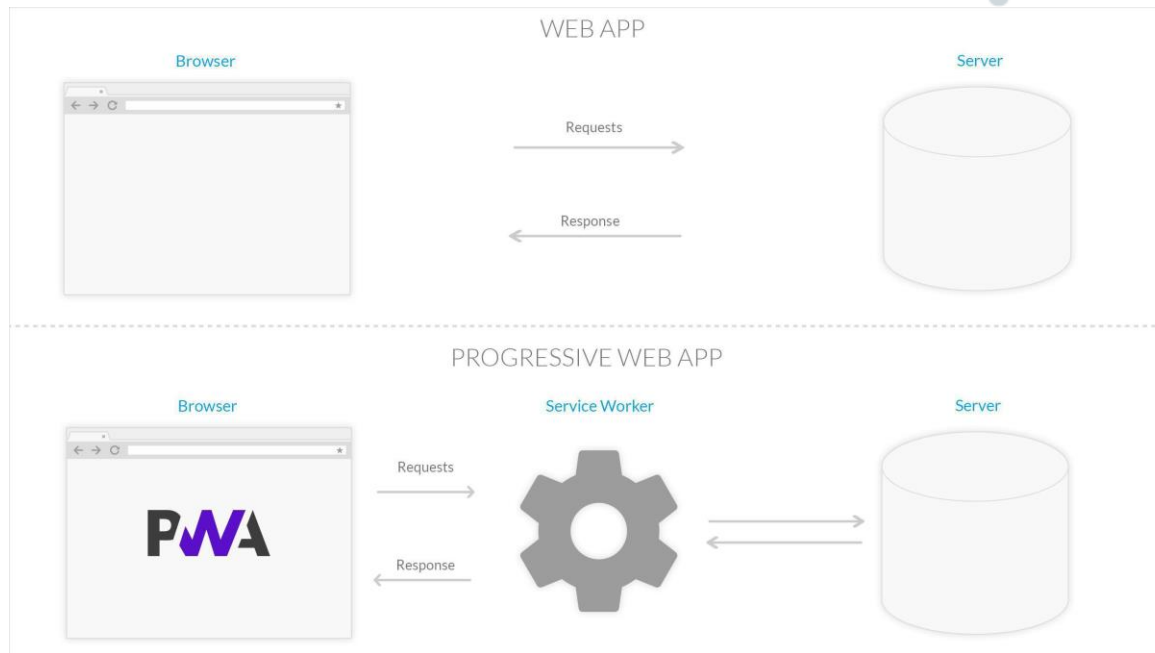
Il termine **Progressive Web App** (PWA, applicazioni web progressive) viene utilizzato per indicare una nuova metodologia per sviluppare software. Diversamente dalle applicazioni tradizionali, le progressive web apps sono un ibrido tra le normali pagine web (o siti web) e le applicazioni mobili. Questo 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](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>

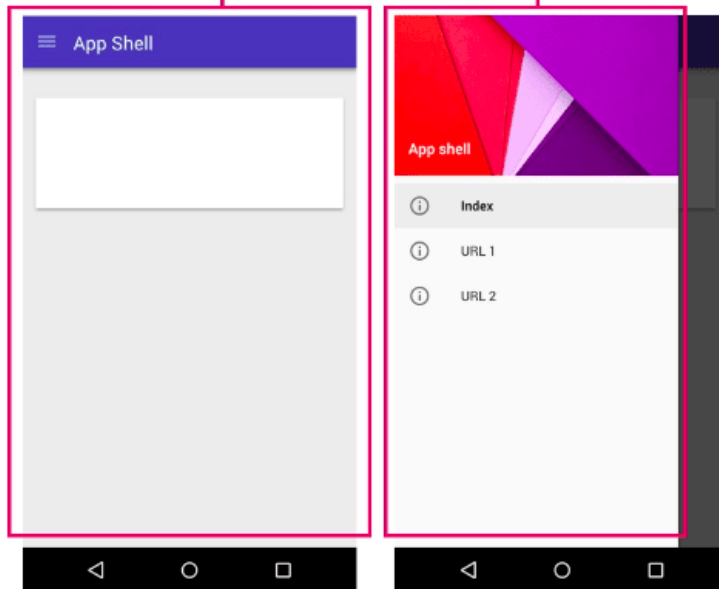
<https://ionicframework.com/pwa>

[https://developers.google.com/web/fundamentals/codelabs/offline#top\\_of\\_page](https://developers.google.com/web/fundamentals/codelabs/offline#top_of_page)

[https://blog.goodbarber.com/it/I-Service-Worker\\_a555.html](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"
}
```

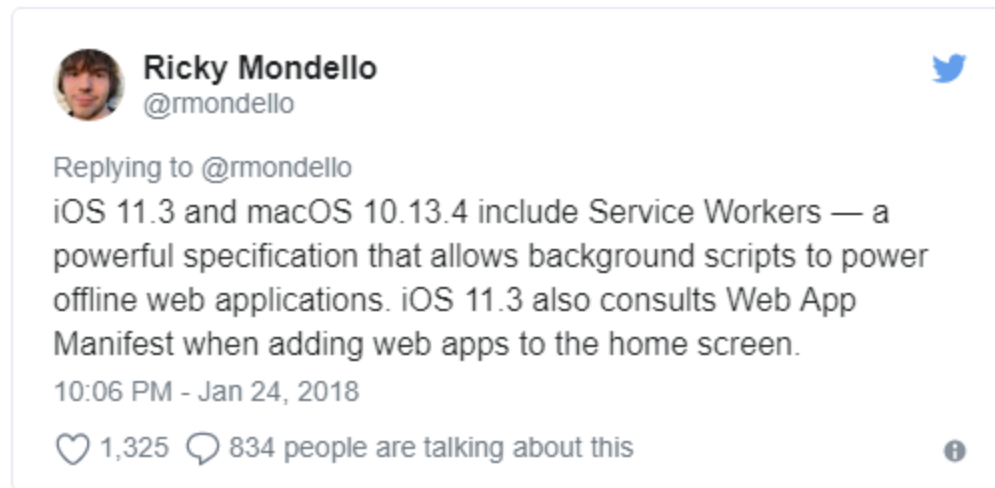


<https://developers.google.com/web/fundamentals/web-app-manifest/>  
<https://developer.mozilla.org/en-US/docs/Web/Manifest>




## PWA: Safari

Update : ios 11.3 supports web app manifest and service worker



A screenshot of a Twitter tweet. The tweet is from Ricky Mondello (@rmondello) and is a reply to another tweet from the same user. 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 has 1,325 likes and 834 replies. The tweet is displayed in a light gray box with a blue Twitter bird icon in the top right corner.

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

## Top PWA examples

### Table of Contents

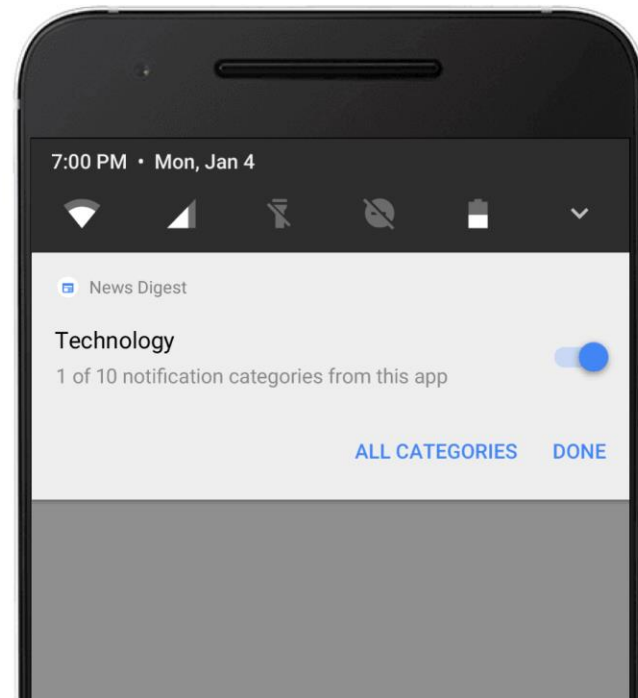
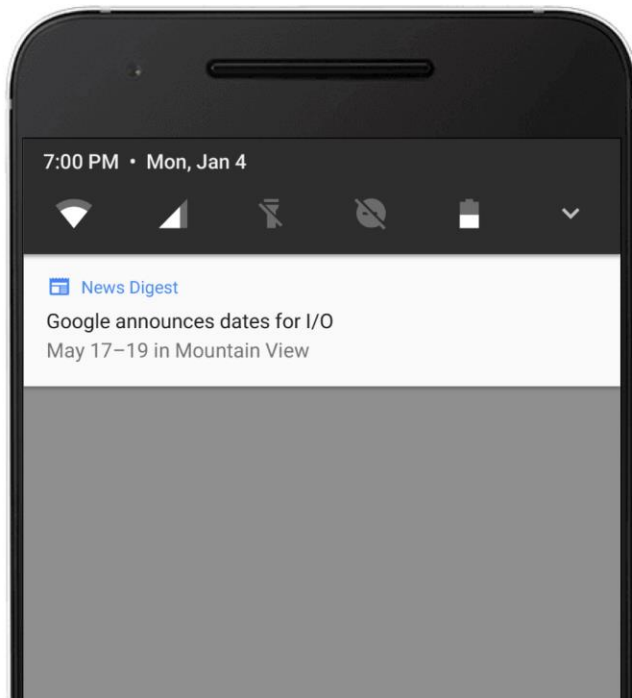
- I. Trivago Hotel Booking
- II. Pinterest
- III. Tinder
- IV. 9Gag
- V. OLX
- VI. Starbucks
- VII. Forbes

<https://pwa.rocks/>

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

## Push Notification



<https://ionicframework.com/docs/native/push>

<https://knowledge.opsview.com/docs/getting-started-with-push>

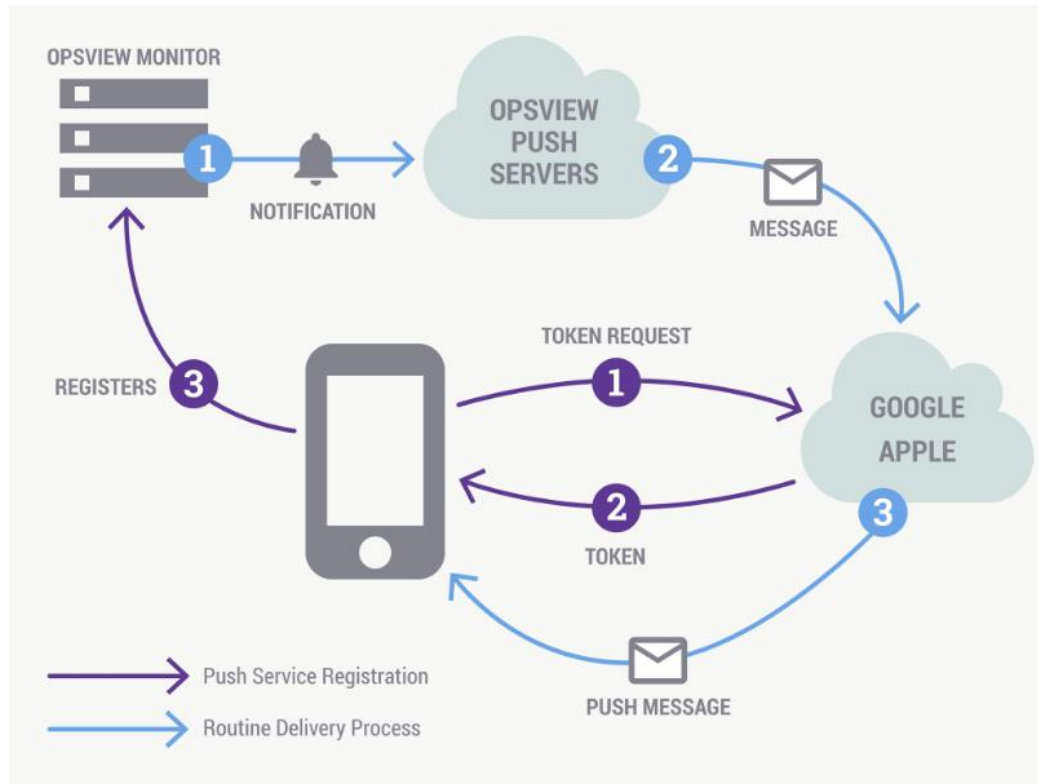
# Push Notification

## Push Notification Terms

- **Notification** – a message displayed to the user outside of the app's normal UI (i.e., the browser)
- **Push Message** – a message sent from the server to the client
- **Push Notification** – a notification created in response to a push message
- **Notifications API** – an interface used to configure and display notifications to the user
- **Push API** – an interface used to subscribe your app to a push service and receive push messages in the service worker
- **Web Push** – an informal term referring to the process or components involved in the process of pushing messages from a server to a client on the web
- **Push Service** – a system for routing push messages from a server to a client. Each browser implements its own push service.
- **Web Push Protocol** – describes how an application server or user agent interacts with a push service

<https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications>

# Push Service



<https://ionicframework.com/docs/native/push>

<https://knowledge.opsview.com/docs/getting-started-with-push>

# Esempio di notifiche con Ionic e FireBase



Completamente free ma:

## Maximum message rate to a single device

You can send up to 240 messages/minute and 5,000 messages/hour to a single device. This high threshold is meant to allow for short term bursts of traffic, such as when users are interacting rapidly over chat. This limit prevents errors in sending logic from inadvertently draining the battery on a device.

**Caution:** Do not routinely send messages near this maximum rate. This could waste end users' resources, and your app may be marked as abusive.

## Upstream message limit

We limit upstream messages at 1,500,000/minute per project to avoid overloading upstream destination servers.

We limit upstream messages per device at 1,000/minute to protect against battery drain from bad app behavior.

<https://www.freecodecamp.org/news/how-to-get-push-notifications-working-with-ionic-4-and-firebase-ad87cc92394e/>

<https://capacitor.ionicframework.com/docs/guides/push-notifications-firebase/>

## Esempio di notifiche con FireBase per servizi backend

```
Node.js  Java  Python  Go  C#  REST

// This registration token comes from the client FCM SDKs.
var registrationToken = 'YOUR_REGISTRATION_TOKEN';

var message = {
  data: {
    score: '850',
    time: '2:45'
  },
  token: registrationToken
};

// Send a message to the device corresponding to the provided
// registration token.
admin.messaging().send(message)
  .then((response) => {
    // Response is a message ID string.
    console.log('Successfully sent message:', response);
  })
  .catch((error) => {
    console.log('Error sending message:', error);
  });
```

<https://firebase.google.com/docs/admin/setup>

<https://firebase.google.com/docs/cloud-messaging/send-message>