



PWA

Progressive Web App





“

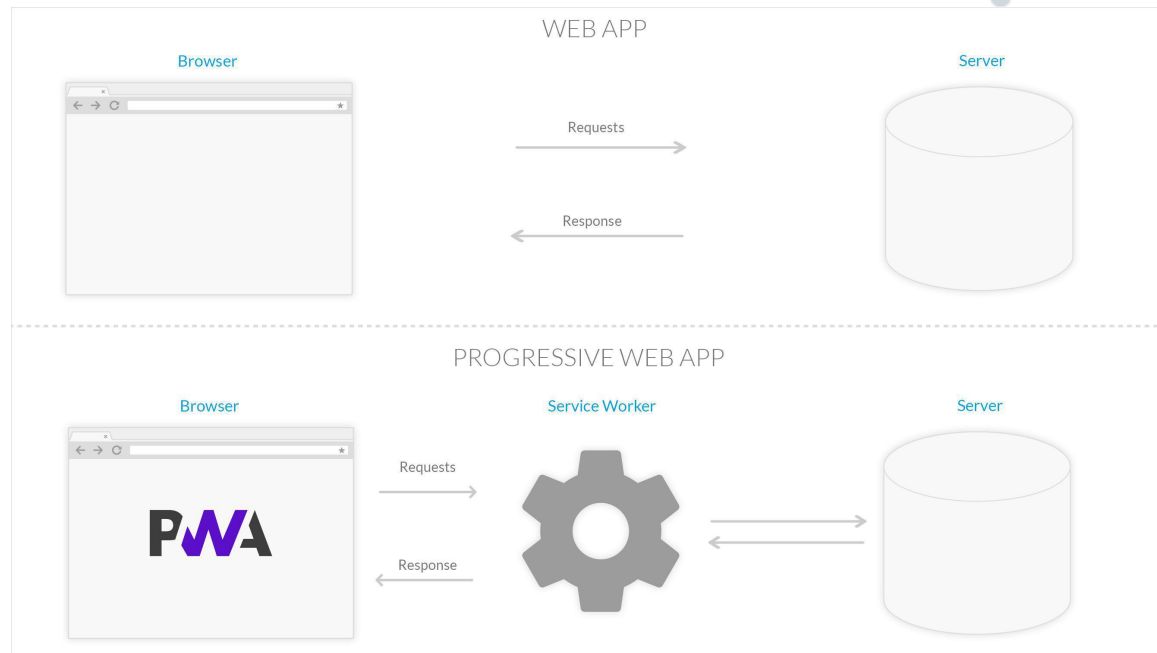
Il termine **Progressive Web App** (PWA, applicazioni web progressive) viene utilizzato per indicare una nuova metodologia per sviluppare software. Diversamente dalle applicazioni tradizionali, le progressive web apps sono un ibrido tra le normali pagine web (o siti web) e le applicazioni mobili. Questo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

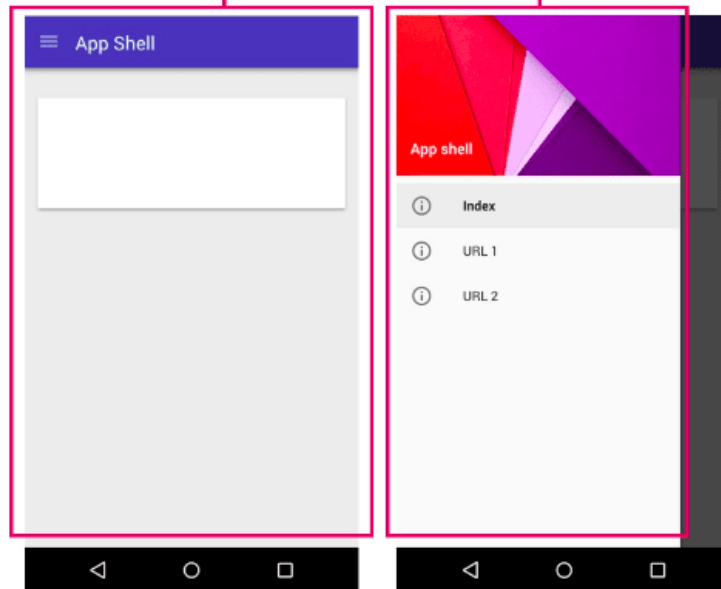
<https://ionicframework.com/pwa>

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

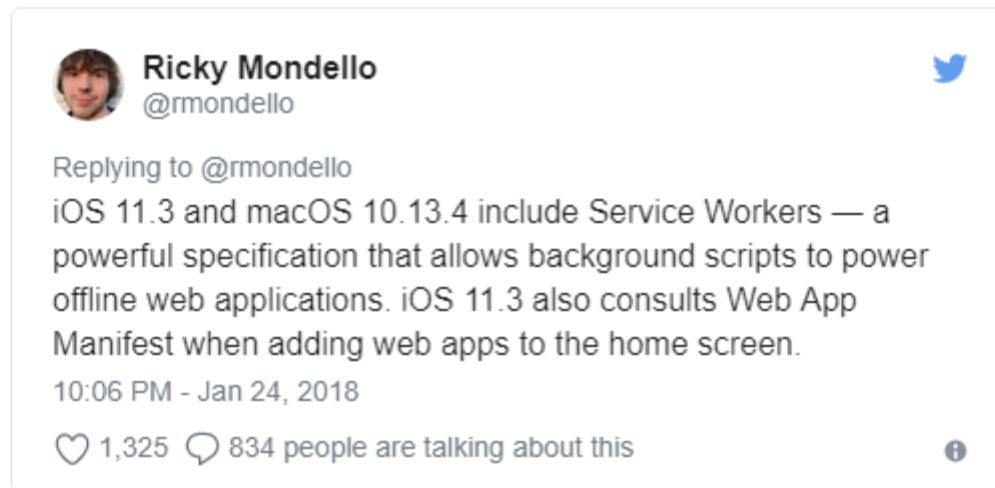
```
{  
  "short_name": "AirHorner",  
  "name": "Kinlan's AirHorner of Infamy",  
  "icons": [  
    {  
      "src": "launcher-icon-1x.png",  
      "type": "image/png",  
      "sizes": "48x48"  
    },  
    {  
      "src": "launcher-icon-2x.png",  
      "type": "image/png",  
      "sizes": "96x96"  
    },  
    {  
      "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 likes 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://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

Ionic





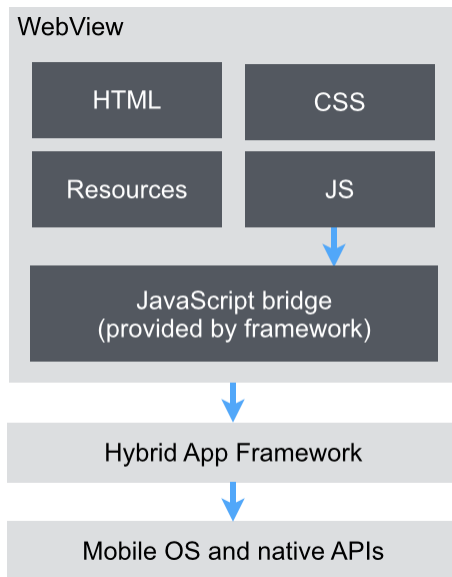
Programma

- Di cosa si tratta?
- Le tecnologie utilizzate
- Setup ambiente
- La struttura del progetto base
- I componenti base
- Alcuni servizi utili
- ...

Di cosa si tratta?

WebView

L'applicazione vive in un browser contenuto nell'app

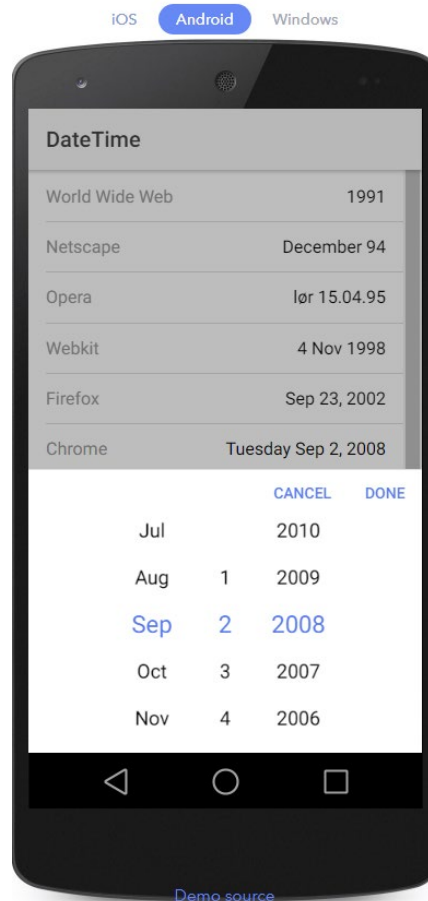
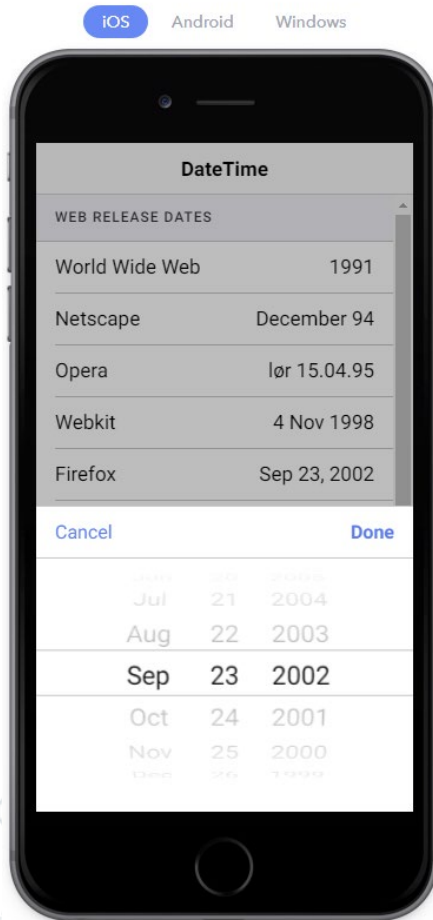


Ionic permette di sviluppare applicazioni mobile ibride sfruttando tecnologie web che tentano di riprodurre il comportamento nativo del sistema

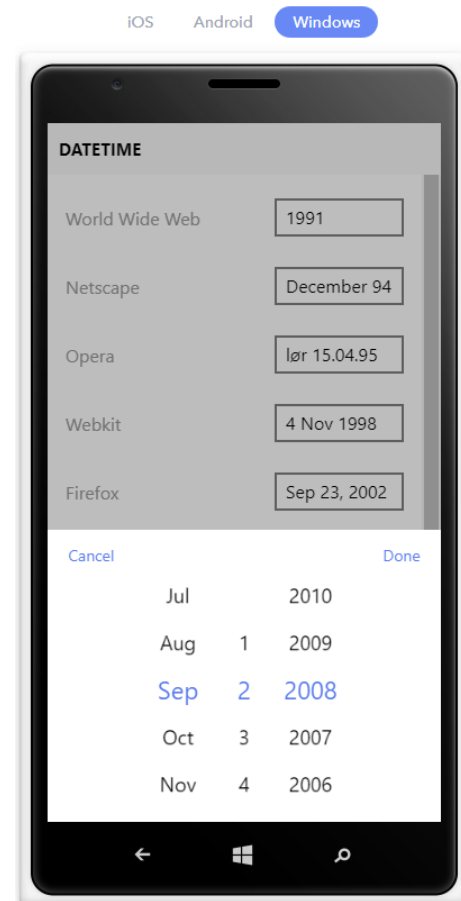
Di cosa si tratta?

Un solo tag html:

```
<ion-datetime displayFormat="MM/DD/YYYY" [(ngModel)]="myDate"></ion-datetime>
```

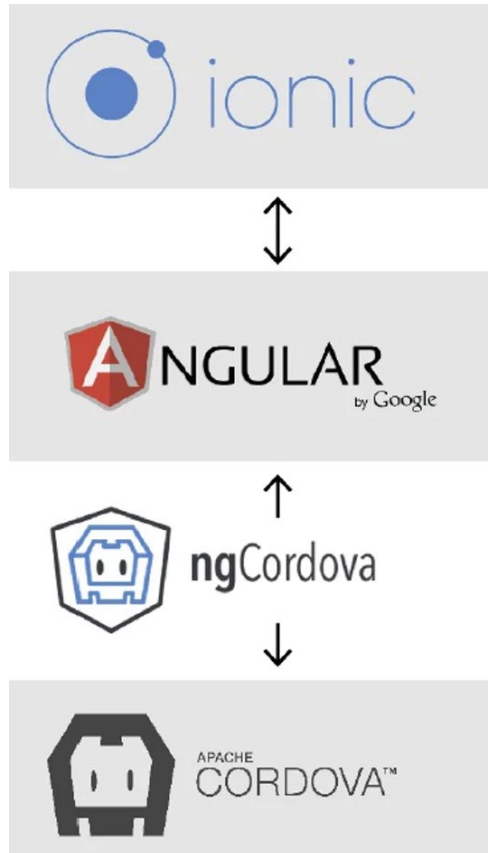


[Demo source](#)



[Demo source](#)

Le tecnologie utilizzate



UI Framework

Framework

Interfacciamento con l'hardware
(dalla versione 3 di ionic si chiama «ionic/native»)

WebView fornita dal progetto Apache Cordova
(Capacitor è il nuovo progetto Ionic per sostituire Cordova)

Le tecnologie utilizzate



TypeScript: linguaggio di programmazione (superset di JavaScript)



HTML5: linguaggio di markup per pagine web



Sass/scss: estensione del css per definire fogli di stile

Setup

- Installare nodejs LTS
- Installare un IDE come VSCODE
- Eseguire: `npm install -g @ionic/cli`
- Eseguire: `ionic start «nomeprogetto»`
- Scegliere «conference» come esempio di app
- Condividere la stessa rete tra notebook e smartphone oppure usare il remote debugging di Chrome
- Entrare nella cartella del progetto ed eseguire: «`ng serve`»

Struttura

Immagini e risorse

app

main

pages

The screenshot shows the Visual Studio Code interface for a project named 'schedule.ts - myProget'. The Explorer view on the left displays the following structure:

- EXPLORER
 - OPEN EDITORS
 - TS schedule.ts src\pages\schedule
 - MYPROGET
 - .github
 - .sourcemaps
 - .tmp
 - node_modules
 - resources
 - src
 - app
 - app.component.ts
 - app.module.ts
 - app.scss
 - app.template.html
 - main.ts
 - assets
 - interfaces
 - pages
 - about
 - about.html
 - about.scss
 - about.ts
 - about-popover
 - about-popover.ts
 - account
 - login
 - login.html
 - login.scss
 - login.ts
 - man

The main editor shows the content of 'TS schedule.ts':

```
1 import { Component, ViewChild } from '@angular/core';
2
3 import { AlertController, App, FabContainer, ItemSliding, List, Mo
4
5 /*
6  To learn how to use third party libs in an
7  Ionic app check out our docs here: http://ionicframework.com/doc
8  */
9 // import moment from 'moment';
10
11 import { ConferenceData } from '../../providers/conference-data';
12 import { UserData } from '../../providers/user-data';
13
14 import { SessionDetailPage } from '../session-detail/session-detail
15 import { ScheduleFilterPage } from '../schedule-filter/schedule-fi
16
17
18 @Component({
19   selector: 'page-schedule',
20   templateUrl: 'schedule.html'
21 })
22 export class SchedulePage {
23   // the list is a child of the schedule page
24   // @ViewChild('scheduleList') gets a reference to the list
25   // with the variable #scheduleList, 'read: List' tells it to ret
26   // the List and not a reference to the element
27   @ViewChild('scheduleList', { read: List }) scheduleList: List;
28
29   dayIndex = 0;
30   queryText = '';
31   segment = 'all';
32   excludeTracks: any = [];
33   shownSessions: any = [];
34   groups: any = [];
```


Pagina

Azione

view

controller

The image shows a Visual Studio Code editor with a project structure on the left and code for an Ionic page in the center. The Explorer sidebar shows a project named 'MYPROGET' with a 'pages' folder containing an 'about' subfolder. The 'about' subfolder contains 'about.html', 'about.scss', and 'about.ts'. The 'about.html' file is selected. The code editor shows the following HTML structure:

```
1 <ion-header>
2   <ion-navbar>
3     <button ion-button menuToggle>
4       <ion-icon name="menu"></ion-icon>
5     </button>
6     <ion-title>About</ion-title>
7   <ion-buttons end>
8     <button ion-button icon-only (click)="presentPopover($event)">
9       <ion-icon name="more"></ion-icon>
10    </button>
11  </ion-buttons>
12 </ion-navbar>
13 </ion-header>
14
15 <ion-content>
16   <div class="about-header">
17     
19   <div padding class="about-info">
20     <h4>Ionic Conference</h4>
21
22   <ion-list no-lines>
23     <ion-item>
24       <ion-icon name="calendar" item-start>
25       <ion-label>Date</ion-label>
26       <ion-datetime displayFormat="MMM D
27     </ion-item>
28
29     <ion-item>
30       <ion-icon name="pin" item-start></
31       <ion-label>Location</ion-label>
32       <ion-select>
33         <ion-option value="madison" sele
34         <ion-option value="austin">Austi
35         <ion-option value="chicago">Chic
36         <ion-option value="seattle">Seatt
37       </ion-select>
38     </ion-item>
```

The code defines an Ionic page with a header, a content area containing an Ionic logo and a heading 'Ionic Conference', and a list of items. Each item has an icon, a label, and a select component. The 'more' button in the header is linked to a 'presentPopover' action. A red arrow points from the 'Azione' label to the click event in the code.

On the right, a preview of the 'About' page is shown. It features a blue header with a hamburger menu icon and the text 'About'. Below the header is the Ionic logo. The main content area displays the 'Ionic Conference' title, followed by a list of items: 'Date' (May 17, 2047) and 'Location' (Madison, WI). A paragraph of text describes the conference. At the bottom, there is a white footer with a grid of icons for 'Schedule', 'Speakers', 'Map', and 'About'.

Componenti per UI

ActionSheetController

AlertController

App

Avatar

Badge

Button

Checkbox

Chip

Col

Config

Content

DateTime

Events

FabButton

FabContainer

FabList

Footer

Grid

Haptic

Header

HideWhen

Icon

<https://ionicframework.com/docs/components/>

Menus

Menu is a side-menu navigation that can be dragged out or toggled to show. The content of a menu will be hidden when the menu is closed.

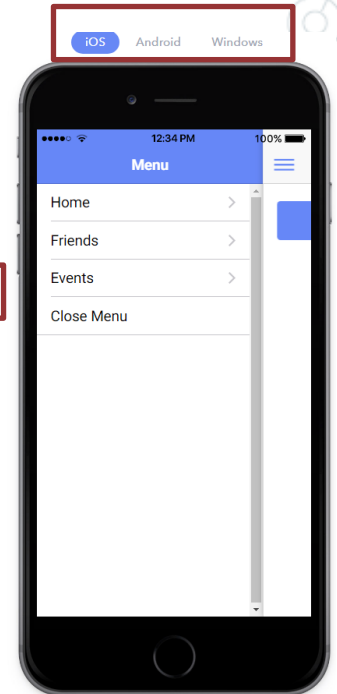
Menu adapts to the appropriate style based on the platform.

For more information, Check out the [API docs](#).

Basic Usage

```
<ion-menu [content]="content">
  <ion-header>
    <ion-toolbar>
      <ion-title>Menu</ion-title>
    </ion-toolbar>
  </ion-header>
  <ion-content>
    <ion-list>
      <button ion-item (click)="openPage(homePage)">
        Home
      </button>
      <button ion-item (click)="openPage(friendsPage)">
        Friends
      </button>
      <button ion-item (click)="openPage(eventsPage)">
        Events
      </button>
      <button ion-item (click)="closeMenu()">
        Close Menu
      </button>
    </ion-list>
  </ion-content>
</ion-menu>
```

[Demo Source](#)



Attiva Windows
Passa a Impostazioni per attivare Windows.

Componente NavController

NavController is the base class for navigation controller components like `Nav` and `Tab`. You use navigation controllers to navigate to `pages` in your app. At a basic level, a navigation controller is an array of pages representing a particular history (of a `Tab` for example). This array can be manipulated to navigate throughout an app by pushing and popping pages or inserting and removing them at arbitrary locations in history.

