

Under the hood

AOT/JIT/Marshaling



Compilatore

AOT

ahead-of-time

La compilazione avviene una sola volta



Nativa

AOT

Ibrida

AOT/JIT

Web (PWA)

JIT

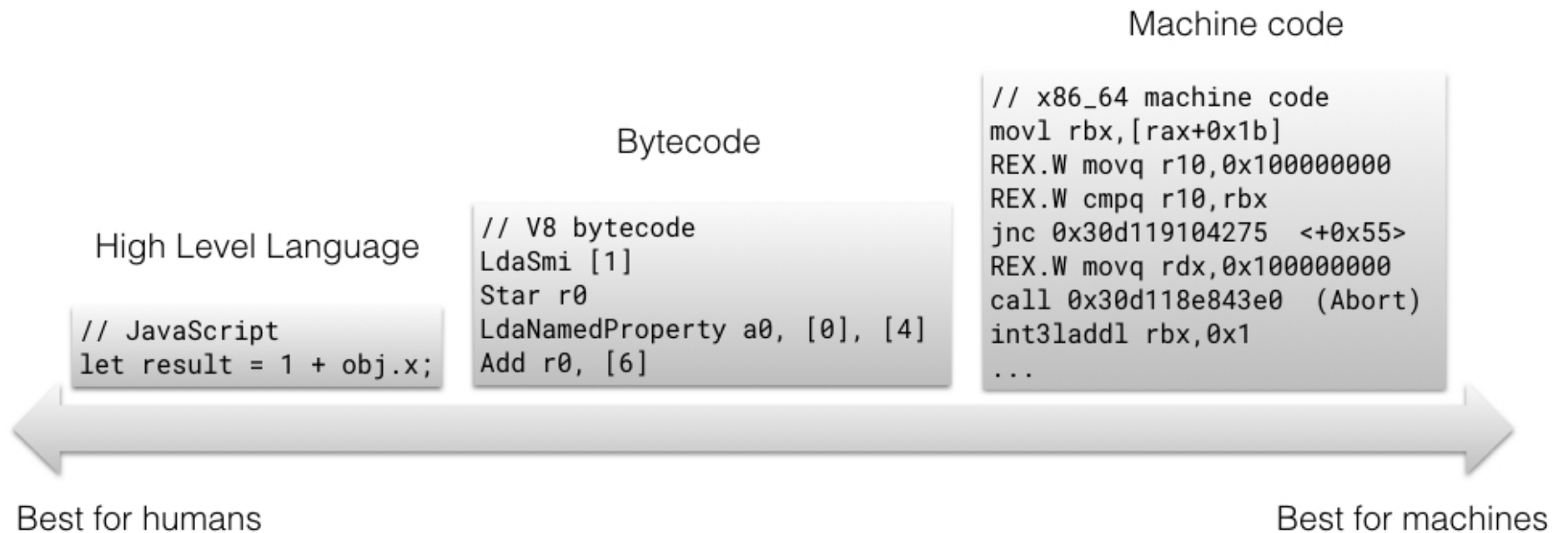
JIT

just-in-time

La compilazione avviene ad ogni avvio

- Android permette il JIT
- iOS non permette la compilazione JIT al di fuori della WKWebView

Javascript

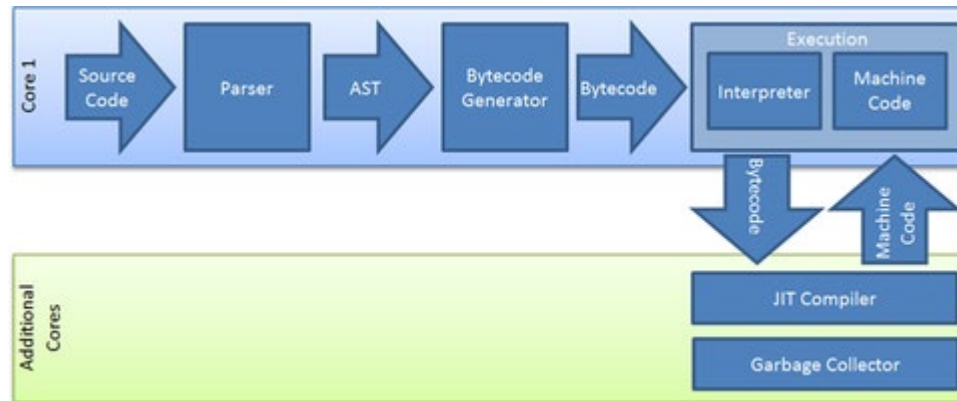


@fhinkel

Compilatore



V8 Engine – Compilazione JIT di un JS





“

Ma come possiamo superare la mancanza di JIT in iOS?

- ▲ No. JavaScriptCore on iOS 7+ won't be able to JIT compile for you, because iOS disallows mapping writable/executable pages of memory as a hard rule, and that's a requirement for JIT. Only
4 MobileSafari.app, Web.app and a handful of other system apps carry an entitlement that allows them to JIT compile. The new WKWebView in iOS 8 is rendered in a separate process that is
▼ allowed to JIT compile, so JavaScript in a WKWebView is faster than a UIWebView or plain JSContext.



share edit flag

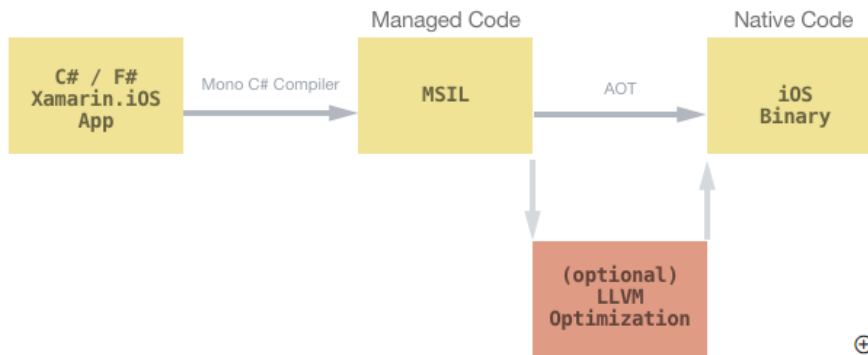
answered Jan 4 '15 at 20:52

Xamarin approach (C#)

AOT

When you compile any Xamarin platform application, the Mono C# (or F#) compiler will run and will compile your C# and F# code into Microsoft Intermediate Language (MSIL). If you are running a Xamarin.Android, a Xamarin.Mac application, or even a Xamarin.iOS application on the simulator, the [.NET Common Language Runtime \(CLR\)](#) compiles the MSIL using a Just in Time (JIT) compiler. At runtime this is compiled into a native code, which can run on the correct architecture for your application.

However, there is a security restriction on iOS, set by Apple, which disallows the execution of dynamically generated code on a device. To ensure that we adhere to these safety protocols, Xamarin.iOS instead uses an Ahead of Time (AOT) compiler to compile the managed code. This produces a native iOS binary, optionally optimized with LLVM for devices, that can be deployed on Apple's ARM-based processor. A rough diagram of how this fits together is illustrated below:



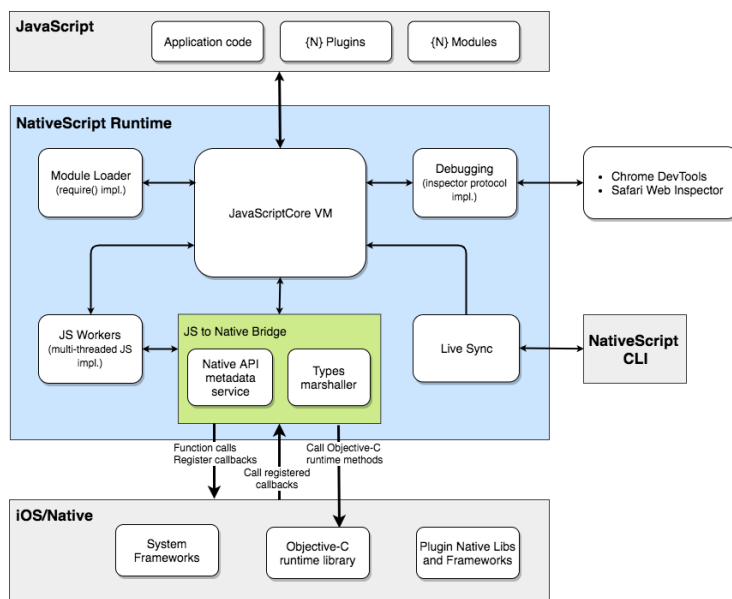
<https://docs.microsoft.com/en-us/xamarin/ios/internals/architecture>

<https://docs.microsoft.com/it-it/xamarin/ios/internals/limitations>

Come superare il limite di iOS ed usare JS?

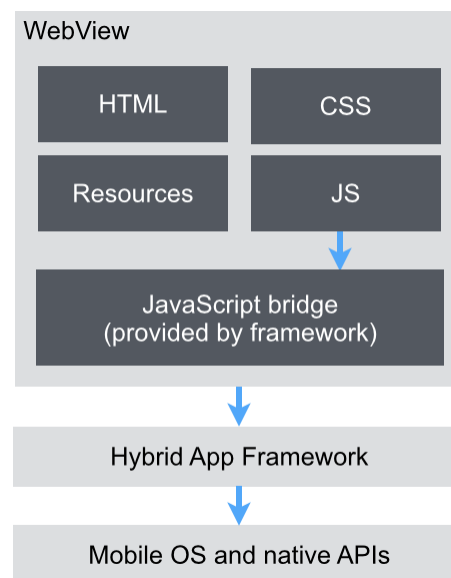
Interprete

L'applicazione viene interpretata a runtime o pre compilata



WebView

L'applicazione vive in un browser contenuto nell'app



<https://docs.nativescript.org/core-concepts/android-runtime/overview>
<https://docs.nativescript.org/core-concepts/ios-runtime/Overview>
<https://www.nativescript.org/blog/the-new-ios-runtime-powered-by-v8>
<https://v8.dev/blog/jitless>

Come superare il limite di iOS ed usare JS?

Interprete

L'applicazione viene interpretata o pre compilata

- Approccio complesso
- Performante
- UI Nativa
- Limiti nella compilazione
- Accesso hardware diretto
- Marshalling

WebView

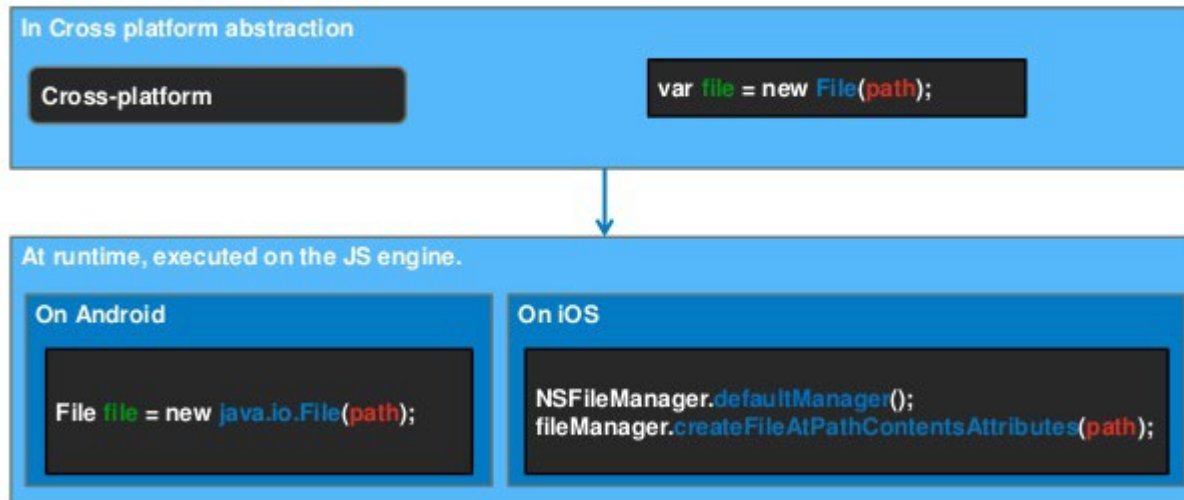
L'applicazione vive in un browser embedded

- Approccio semplice
- Lenta
- UI Web
- Nessun limite (JIT presente)
- Limiti nell'accesso hardware

Marshalling

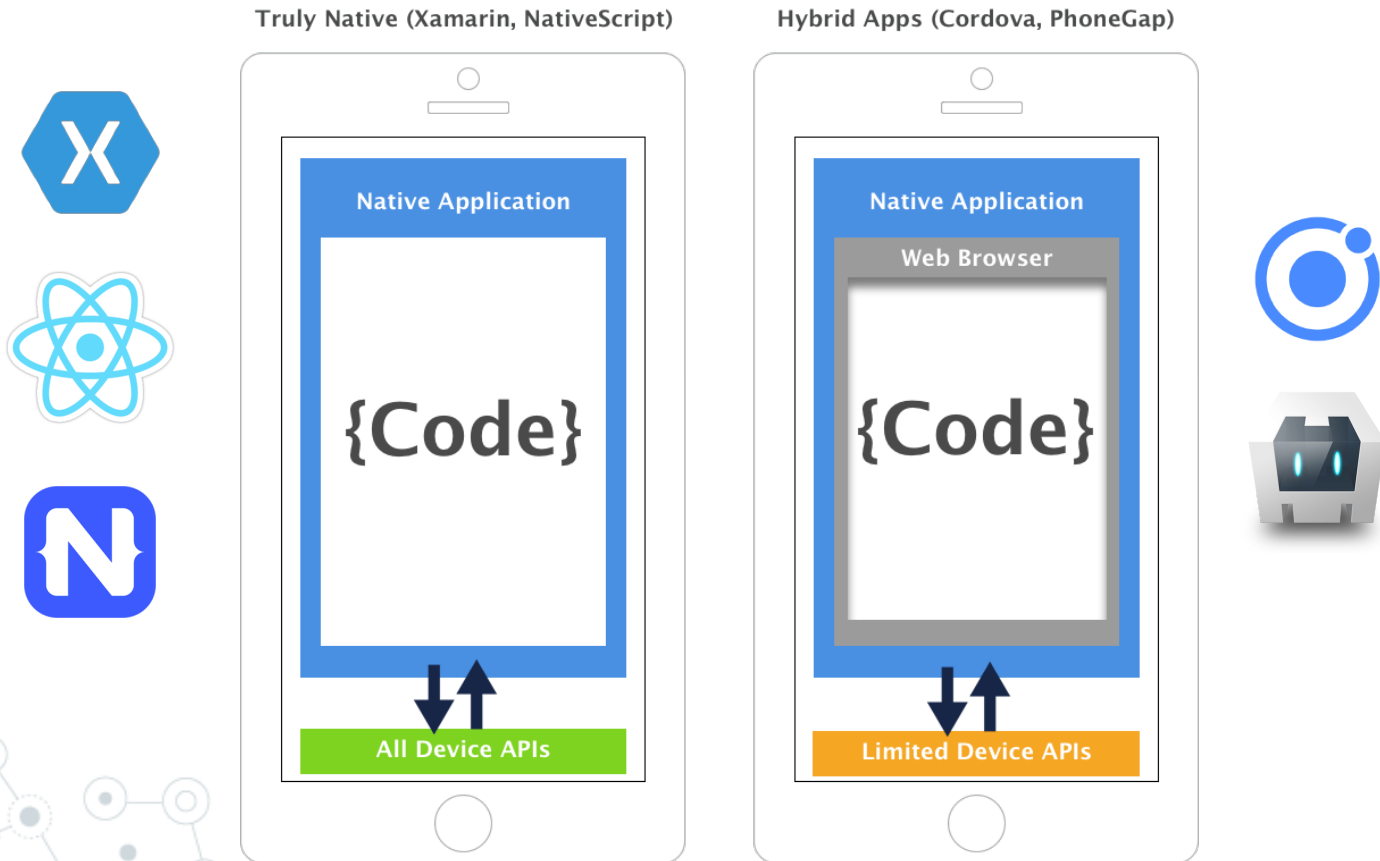
<https://docs.nativescript.org/runtimes/android/marshalling/overview>
<https://docs.nativescript.org/runtimes/ios/marshalling/Marshalling-Overview>

Cross-platform API



Differenze sostanziali

<https://www.nativescript.org/blog/nativescript-and-xamarin>



Security

resources



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

<https://hunter.how/>

<https://www.zoomeye.hk/>

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

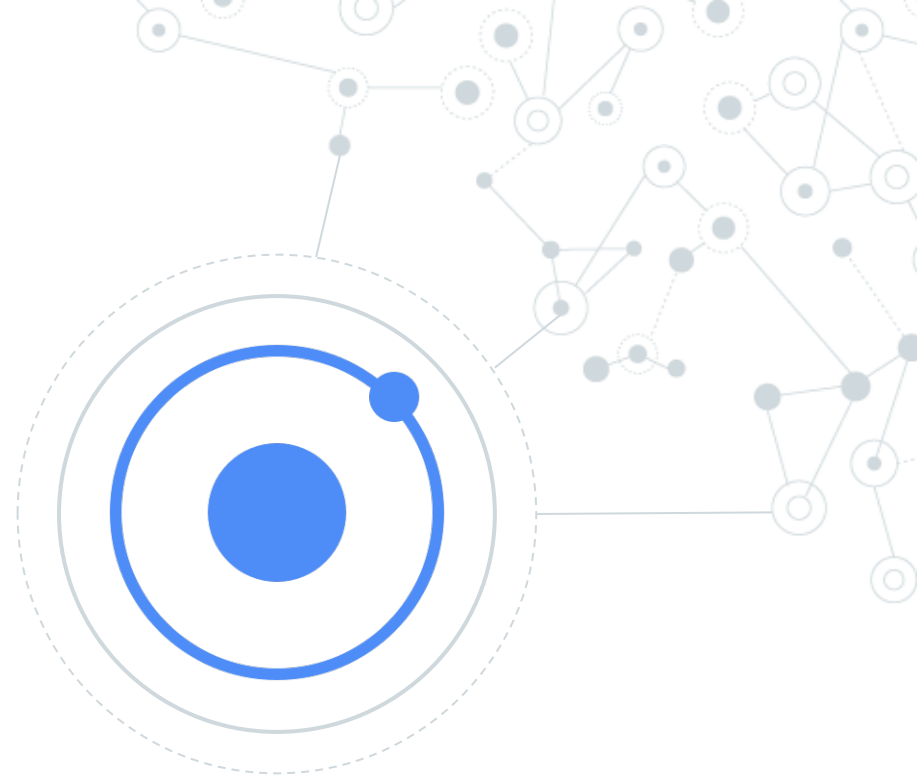
<https://www.cvedetails.com/>

<https://worldofvnc.net/>

<https://haveibeenpwned.com/>

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

Ionic





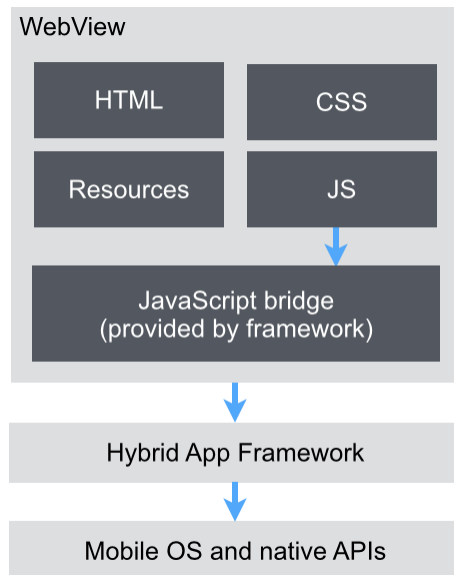
Programma

- 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

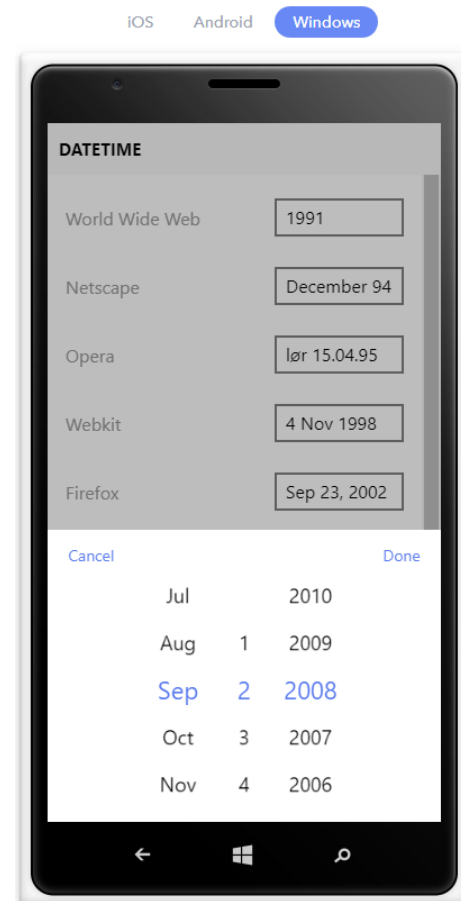
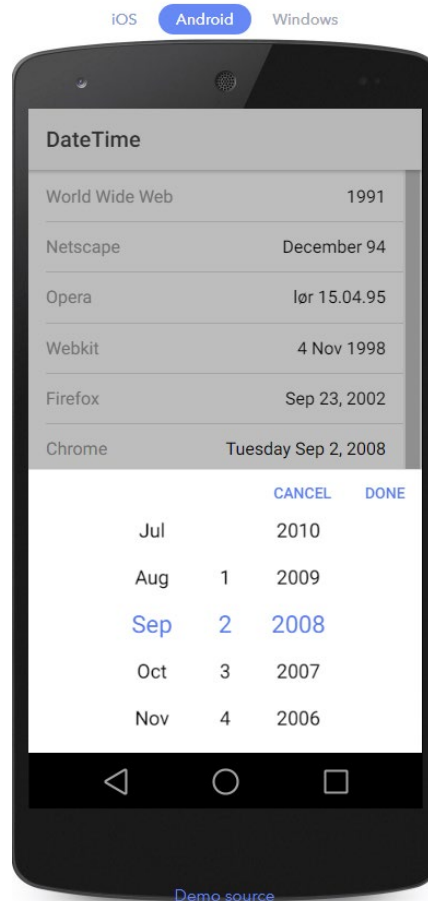
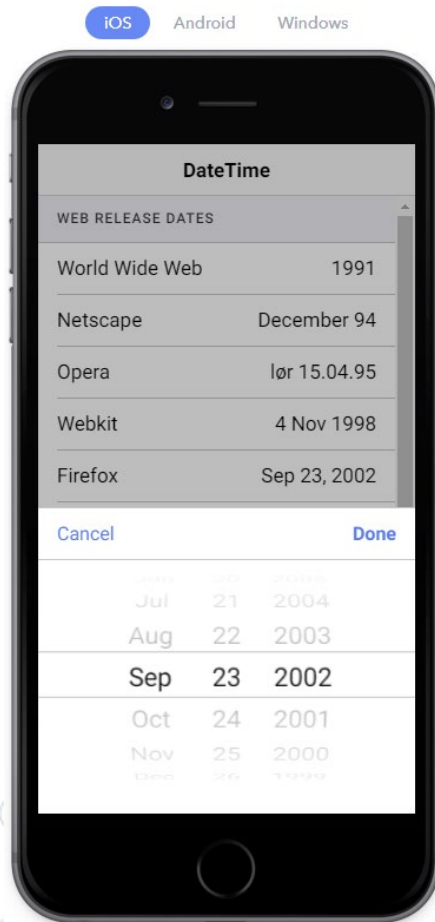


Ionic permette di sviluppare applicazioni mobile ibride sfruttando tecnologie web che tentano di riprodurre il comportamento nativo del sistema

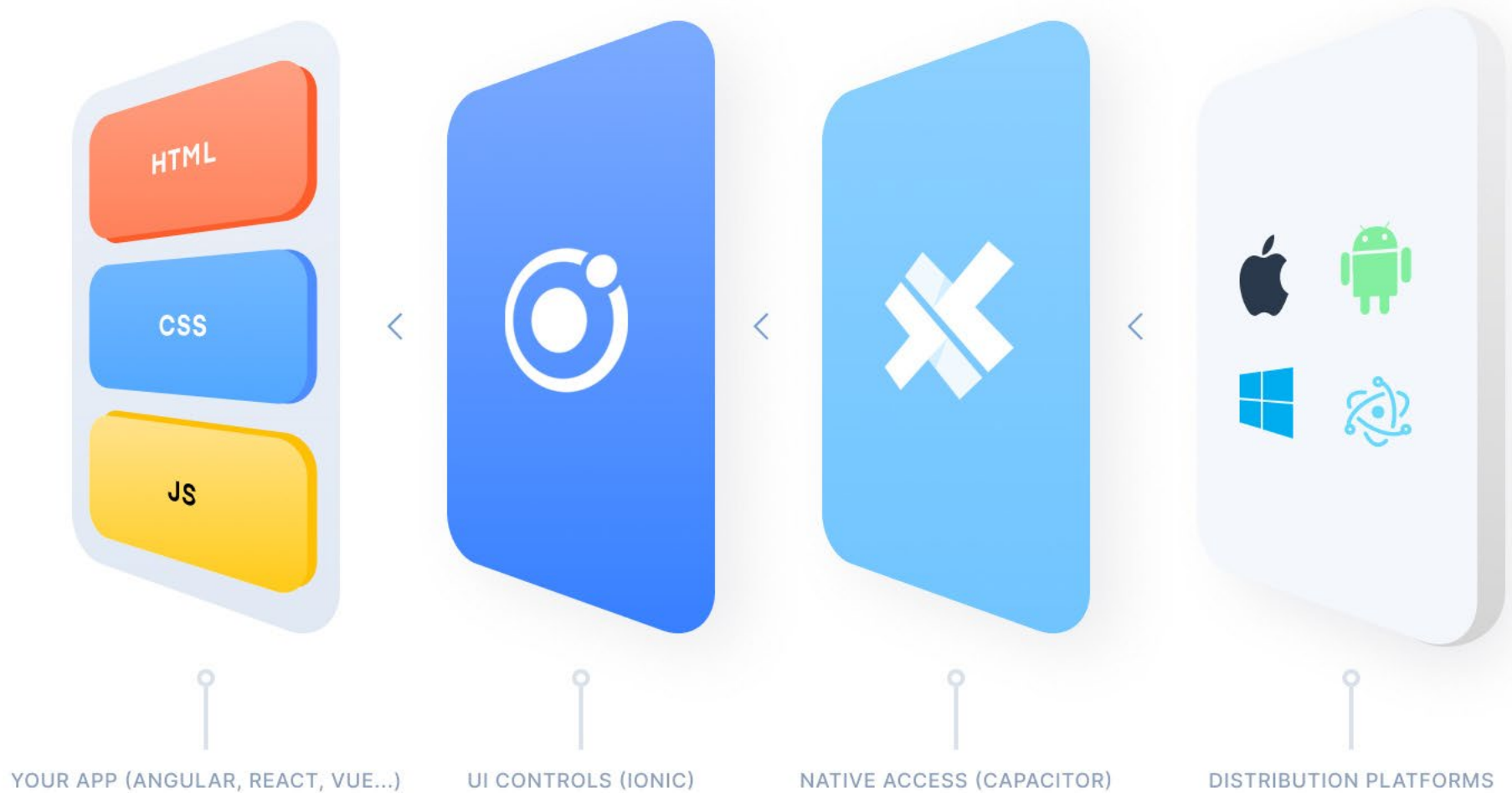
Di cosa si tratta?

Un solo tag html:

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



Le tecnologie utilizzate



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»

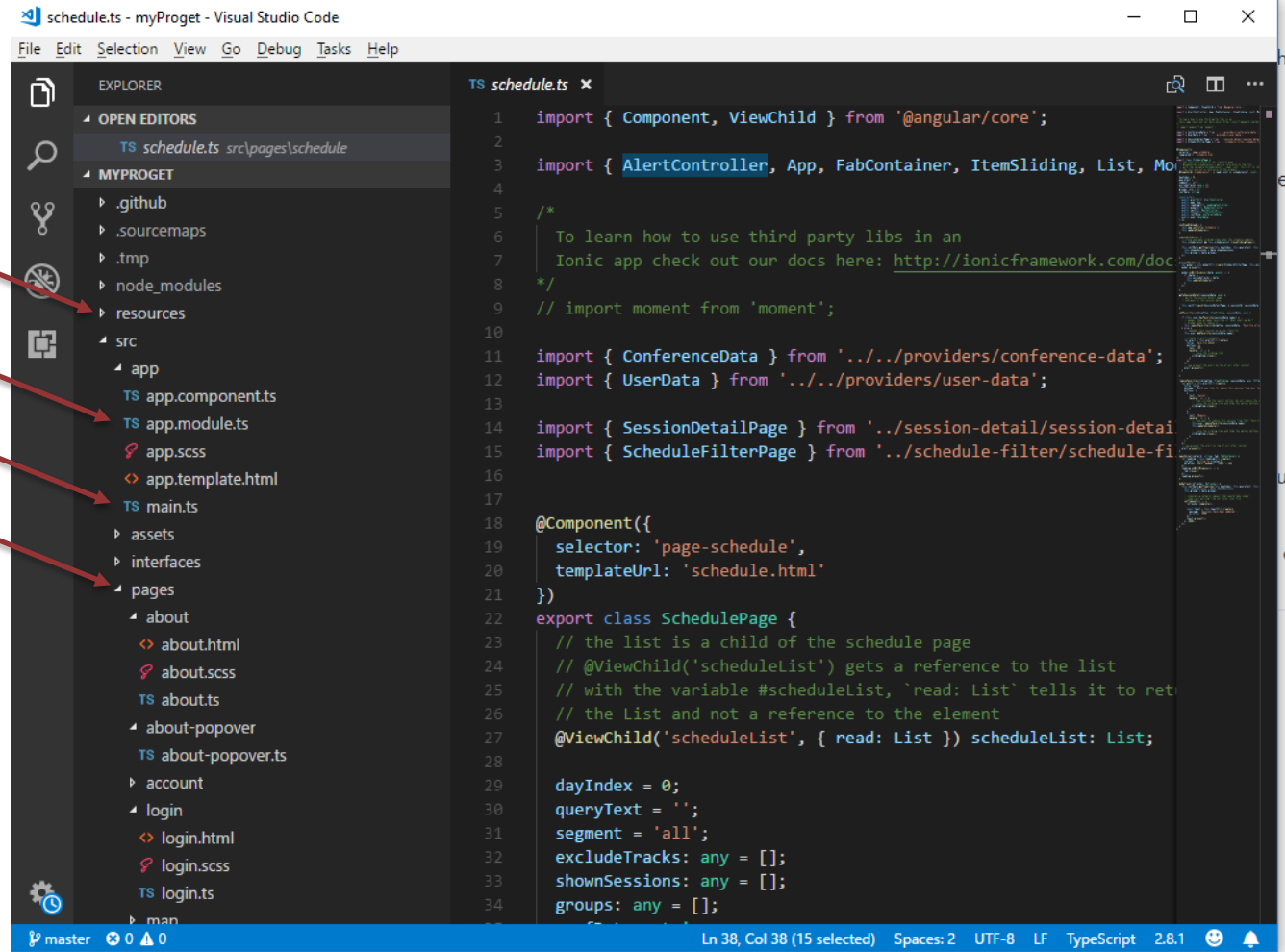
Struttura

Immagini e risorse

app

main

pages



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the editor on the right. The Explorer sidebar displays the project structure for 'schedule.ts - myProget'. The 'src' directory is expanded, showing subdirectories like 'app', 'assets', 'interfaces', and 'pages'. The 'pages' directory is further expanded, showing files like 'about.html', 'about.scss', 'about.ts', 'about-popover.ts', 'account', 'login.html', 'login.scss', and 'login.ts'. The 'main.ts' file is highlighted in the 'src' directory. The editor shows the content of 'main.ts', which includes imports from '@angular/core', '@ionic/angular', and various providers. It also shows the '@Component' decorator and the 'SchedulePage' class.

```
1 import { Component, ViewChild } from '@angular/core';
2
3 import { alertController, App, FabContainer, ItemSliding, List, ModalController } from '@ionic/angular';
4
5 /**
6  * To learn how to use third party libs in an
7  * Ionic app check out our docs here: http://ionicframework.com/docs/v3/tutorial/native-libs/
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-filter';
16
17 @Component({
18   selector: 'page-schedule',
19   templateUrl: 'schedule.html'
20 })
21 export class SchedulePage {
22   // the list is a child of the schedule page
23   // @ViewChild('scheduleList') gets a reference to the list
24   // with the variable #scheduleList, 'read: List' tells it to return
25   // the List and not a reference to the element
26   @ViewChild('scheduleList', { read: List }) scheduleList: List;
27
28   dayIndex = 0;
29   queryText = '';
30   segment = 'all';
31   excludeTracks: any = [];
32   shownSessions: any = [];
33   groups: any = [];
```

Pagina

Azione

view

controller

The image shows a VS Code interface with the following components:

- EXPLORER:** Displays the project structure. The file `about.html` is selected under the `pages/about` directory.
- CODE EDITOR:** Shows the content of `about.html`. The code includes an `<ion-header>` with a menu toggle button and an `<ion-content>` with a header, date, location, and a list of items.
- MOBILE APP PREVIEW:** A small window on the right shows a mobile app interface. It has a blue header with the text "About" and a hamburger menu icon. Below the header is a white card with the Ionic logo and the text "Ionic Conference". The card also displays the date "May 17, 2047" and the location "Madison, WI". At the bottom of the preview, there is a navigation bar with icons for "Schedule", "Speakers", "Map", and "About".

```
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-navbar>
12 </ion-header>
13
14 <ion-content>
15   <div class="about-header">
16     
18   <div padding class="about-info">
19     <h4>Ionic Conference</h4>
20
21   <ion-list no-lines>
22     <ion-item>
23       <ion-icon name="calendar" item-start>
24       <ion-label>Date</ion-label>
25       <ion-datetime displayFormat="MMM D
26     </ion-item>
27
28     <ion-item>
29       <ion-icon name="pin" item-start></
30       <ion-label>Location</ion-label>
31       <ion-select>
32         <ion-option value="madison" sele
33         <ion-option value="austin">Austi
34         <ion-option value="chicago">Chic
35         <ion-option value="seattle">Seatt
36       </ion-select>
37     </ion-item>
38   </ion-list>
39 </ion-content>
```

ference

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.

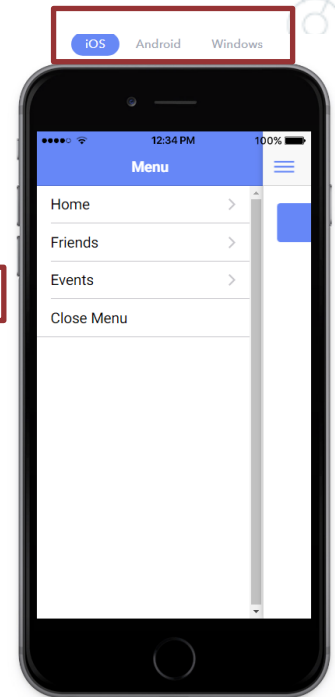
Menu adapts to the appropriate style based on the platform.

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>
```

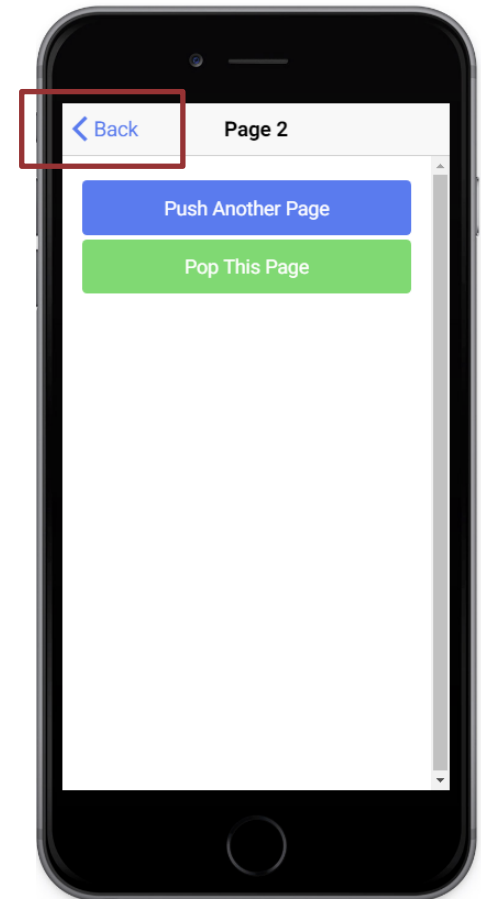
[Demo Source](#)



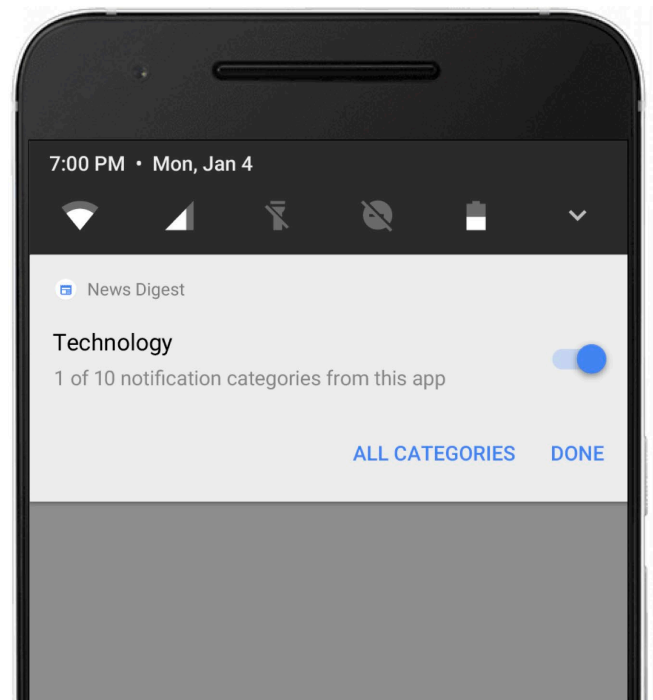
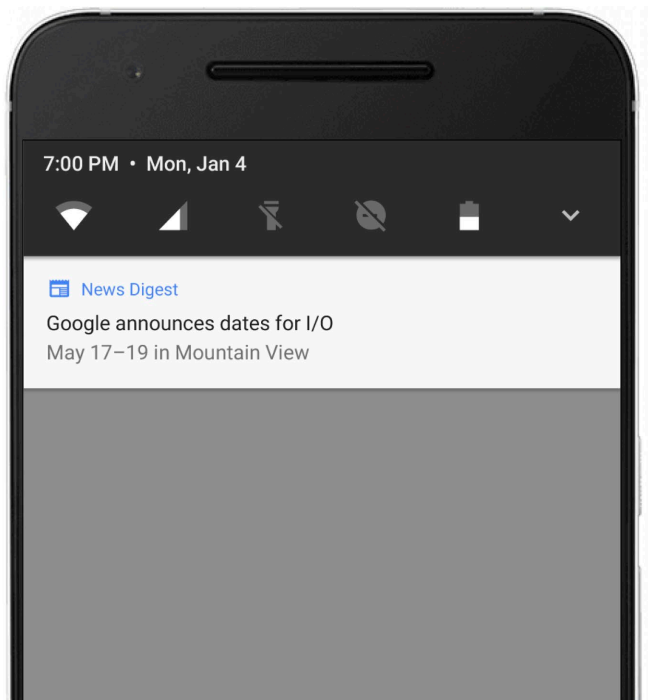
Attiva Windows
Passa a Impostazioni per attivare Windows.

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.



Push Notification



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

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



Progressive Web App





“

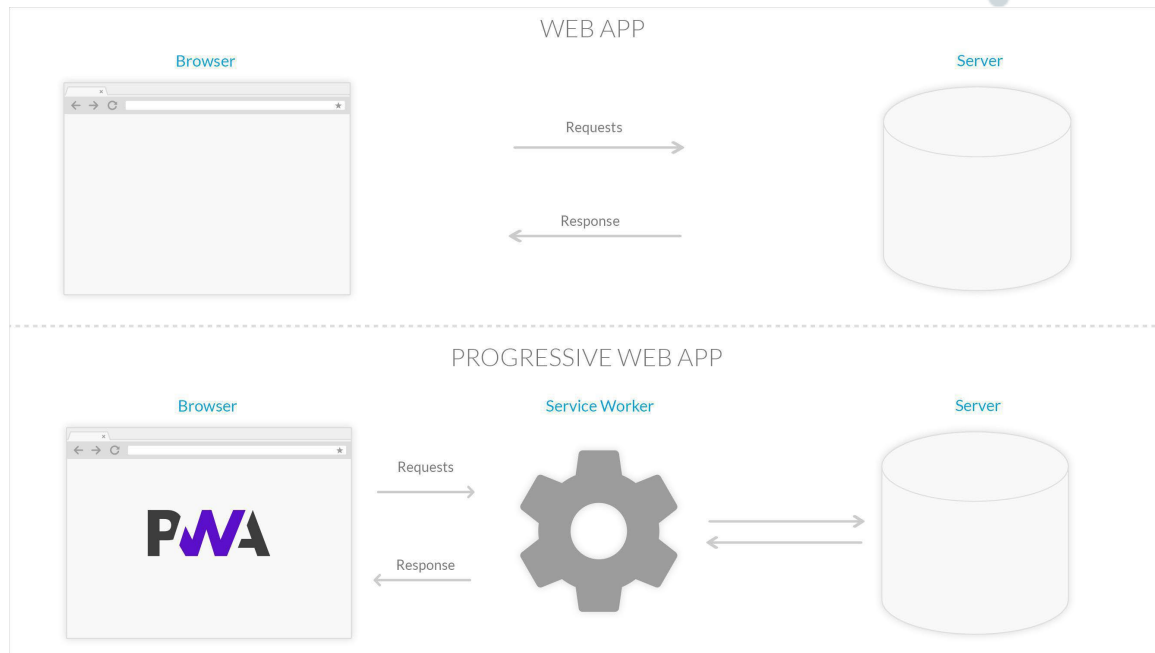
Il termine **Progressive Web App** (PWA, applicazioni web progressive) viene utilizzato per indicare una nuova metodologia per sviluppare software. Diversamente dalle applicazioni tradizionali, le progressive web apps sono un ibrido tra le normali pagine web (o siti web) e le applicazioni mobili. Questo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

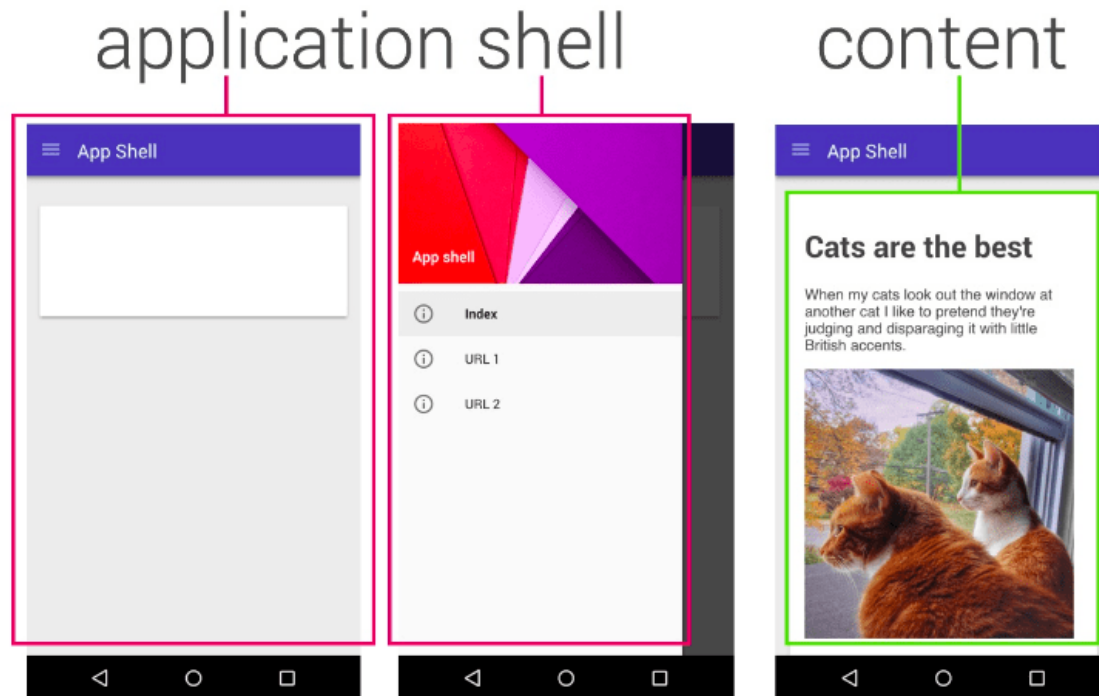
<https://developers.google.com/web/tools/workbox>

<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



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

Dynamic content then populates the view

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

PWA: Manifest

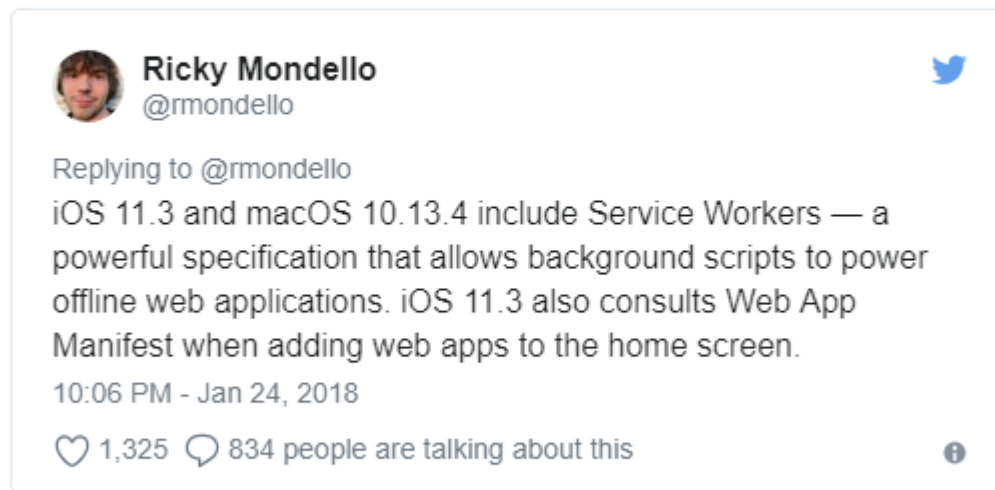
```
{
  "short_name": "AirHorner",
  "name": "Kinlan's AirHorner of Infamy",
  "icons": [
    {
      "src": "launcher-icon-1x.png",
      "type": "image/png",
      "sizes": "48x48"
    },
    {
      "src": "launcher-icon-2x.png",
      "type": "image/png",
      "sizes": "96x96"
    },
    {
    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



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