

Under the hood

AOT/JIT/Marshaling



Compilatore

AOT

ahead-of-time

La compilazione avviene una sola volta



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

Nativa

Ibrida

Web (PWA)

AOT

AOT/JIT

JIT

Javascript



Machine code

```
// x86_64 machine code  
movl rbx,[rax+0x1b]  
REX.W movq r10,0x100000000  
REX.W cmpq r10,rbx  
jnc 0x30d119104275 <+0x55>  
REX.W movq rdx,0x100000000  
call 0x30d118e843e0 (Abort)  
int3laddl rbx,0x1  
...
```

Bytecode

```
// V8 bytecode  
LdaSmi [1]  
Star r0  
LdaNamedProperty a0, [0], [4]  
Add r0, [6]
```

High Level Language

```
// JavaScript  
let result = 1 + obj.x;
```



Best for humans

Best for machines



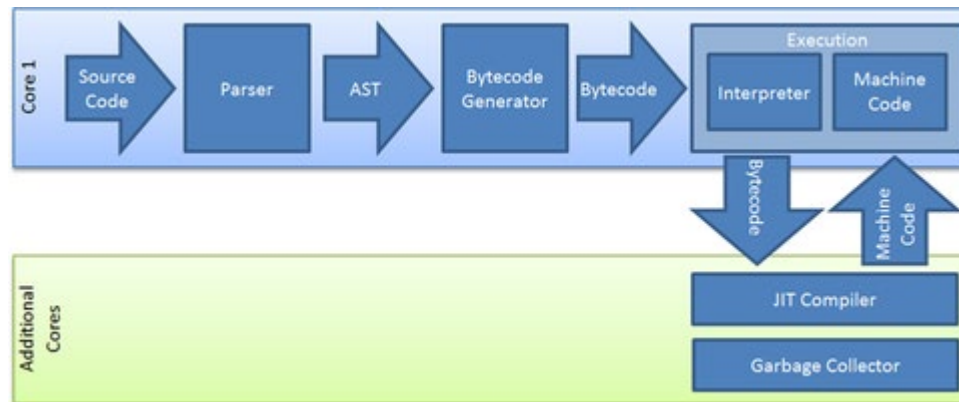
@finkel



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 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

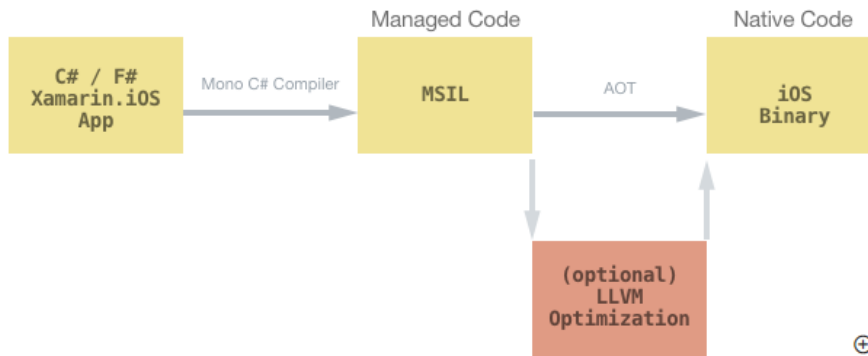
answered Jan 4 '15 at 20:52

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:



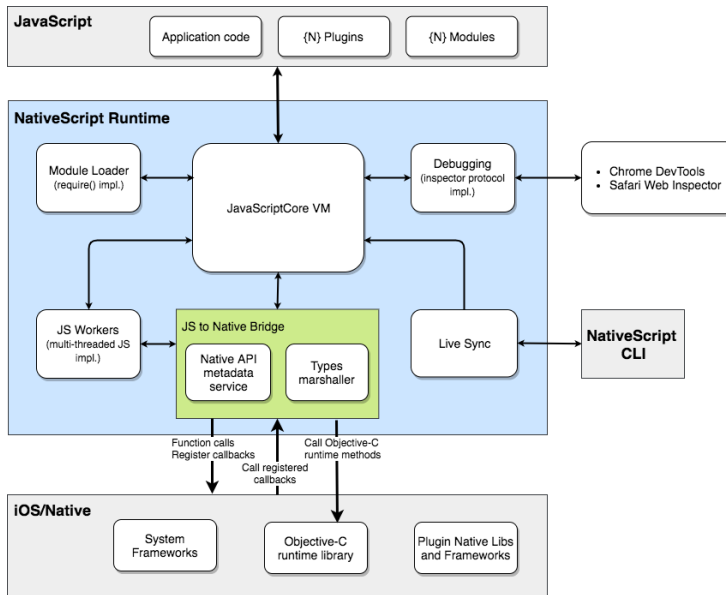
<https://docs.microsoft.com/en-us/xamarin/ios/internals/architecture>

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

Come superare il limite di iOS ed usare JS?

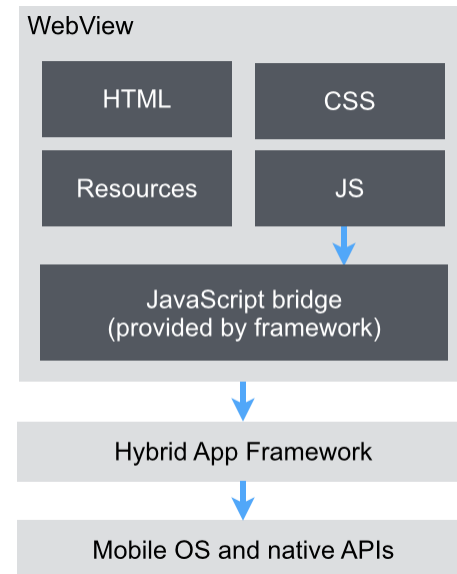
Interprete

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

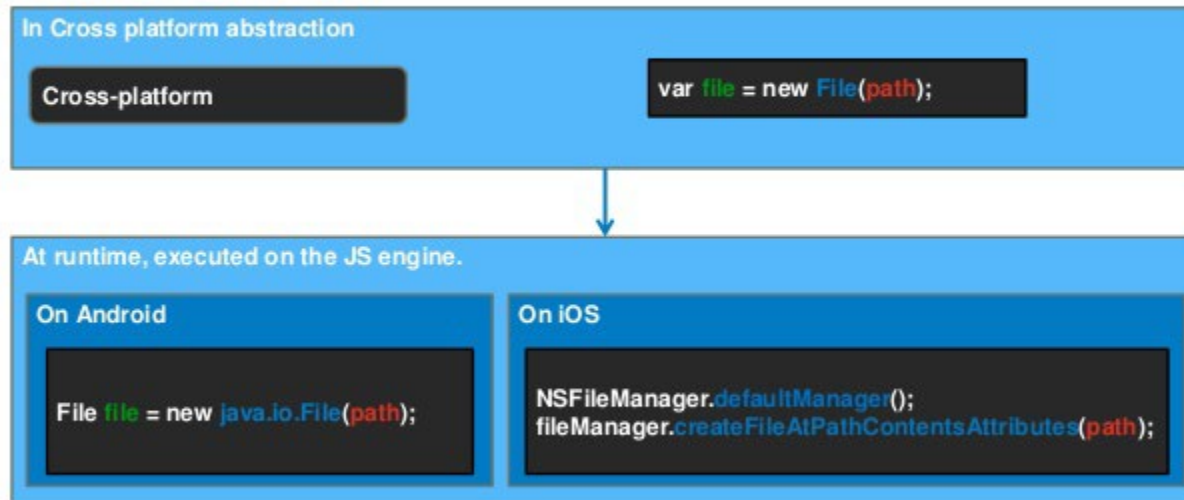
L'applicazione vive in un browser embedded

- 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

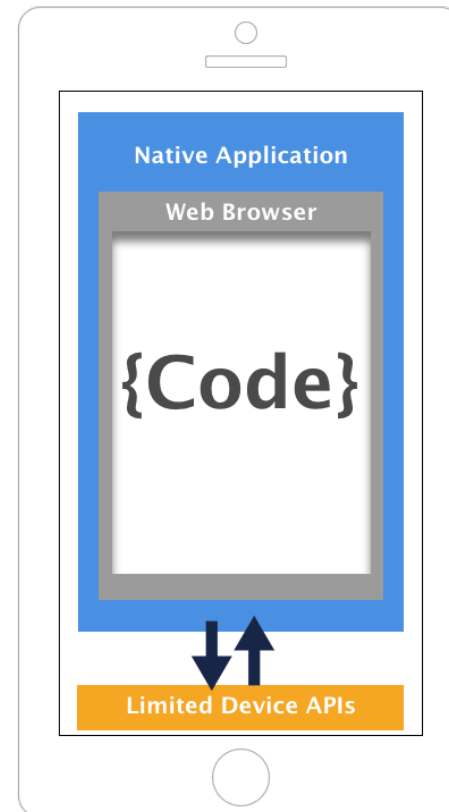
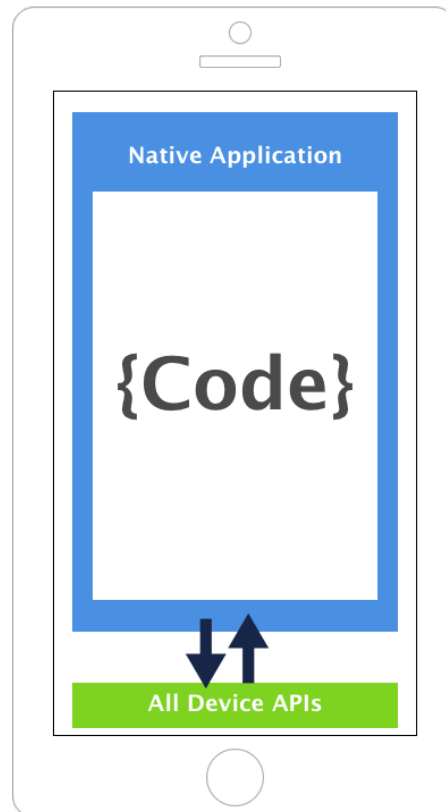


Differenze sostanziali

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

Truly Native (Xamarin, NativeScript)

Hybrid Apps (Cordova, PhoneGap)



Ionic





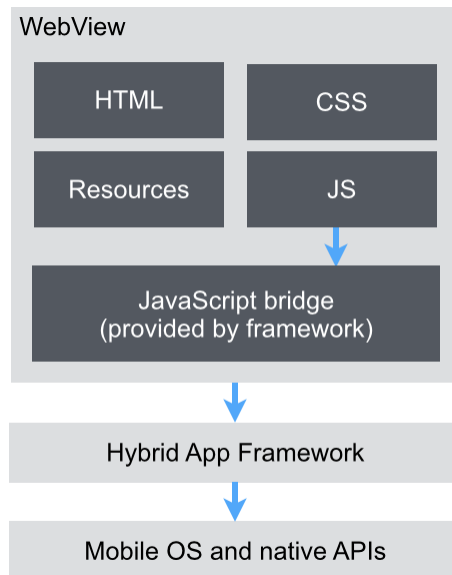
Programma

- 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

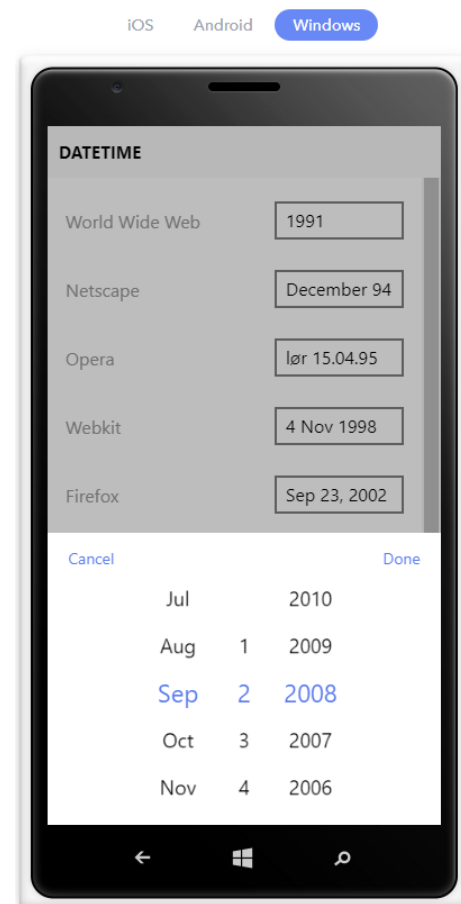
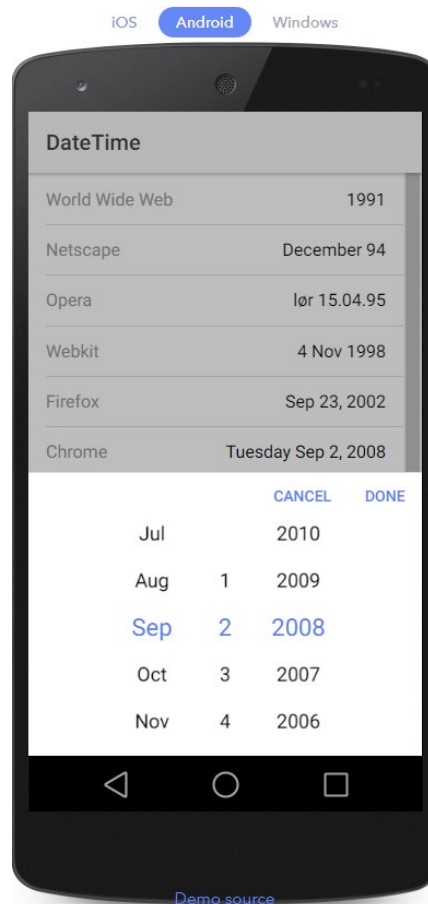
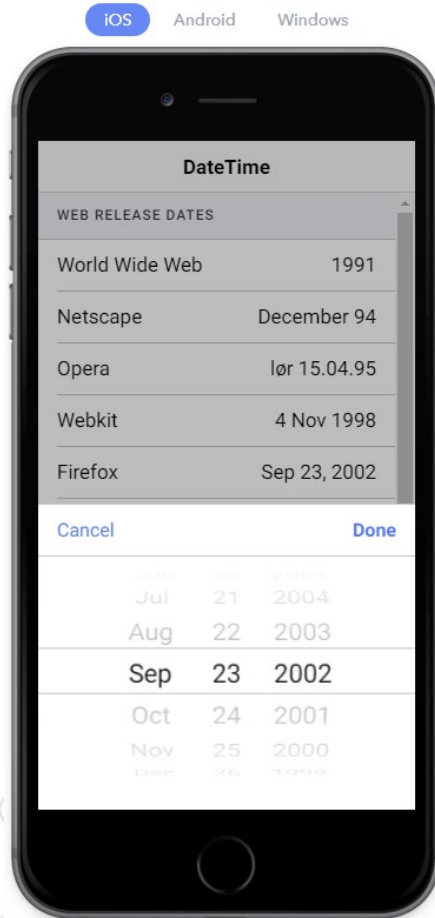


Ionic permette di sviluppare applicazioni mobile ibride sfruttando tecnologie web che tentano di riprodurre il comportamento nativo del sistema

Di cosa si tratta?

Un solo tag html:

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



Le tecnologie utilizzate



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

schedule.ts - myProget - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXPLORER

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

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 = [];
```

Ln 38, Col 38 (15 selected) Spaces: 2 UTF-8 LF TypeScript 2.8.1

Pagina

Azione

view

controller

The image shows a development environment (VS Code) with the following components:

- EXPLORER:** Shows the project structure. The file `about.html` is selected under the `pages/about` directory.
- EDITOR:** Displays the HTML code for `about.html`. The code includes an `<ion-header>` with a `<ion-navbar>` containing a `<button>` with an `<ion-icon>` and a `<ion-title>`. The `<ion-content>` contains a header with the Ionic logo and a list of items with icons and labels.
- MOBILE PREVIEW:** Shows the rendered mobile application. The page title is "About". It features the Ionic logo and a list of items with icons and labels, matching the code in the editor.

Annotations in the image:

- A red arrow labeled "view" points to the `about.html` file in the Explorer.
- A red arrow labeled "Azione" points to the `<button>` element in the code editor.
- A red arrow labeled "controller" points to the `about.ts` file in the Explorer.

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.

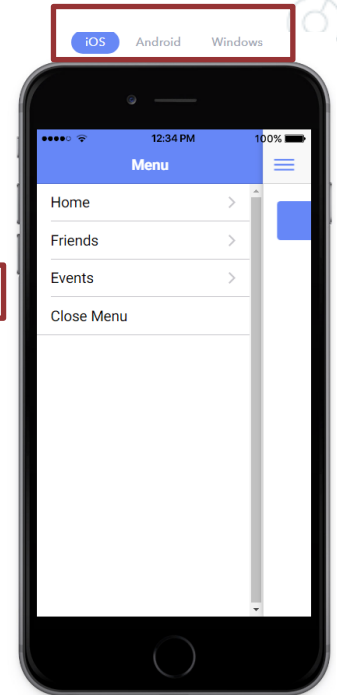
Menu adapts to the appropriate style based on the platform.

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.

