

[Apache Cordova](#) یک فریمورک سورس باز برای ساخت اپلیکیشن‌های چند سکویی موبایل (cross platform) با استفاده از Html5 می‌باشد.

طی چند مقاله، با استفاده کردن از این فریمورک در VS آشنا خواهیم شد. هدف خالقان Cordova یافتن یک راه ساده برای تولید اپلیکیشن‌های چند سکویی موبایل بود که برای رسیدن به این هدف تصمیم گرفتند از تکنولوژی‌های بومی (native) و تکنولوژی‌های وب استفاده کنند. به این نوع از اپلیکیشن‌های موبایل، Hybrid Application می‌گویند. Cordova دارای قابلیت‌های بومی بالایی است و مهم‌تر اینکه به طور طبیعی توسط مرورگرها پشتیبانی می‌شود. بعد از تولد Corodva، این فریمورک تبدیل شده است به بهترین روش تولید اپلیکیشن‌هایی که بر روی چند نوع پلتفرم کار می‌کنند. پیشتر محدودیتی که وجود داشت شامل این بود که اپلیکیشن‌های موبایل، به چیزهایی بیشتر از HTML و مرورگرهای وب، نیاز داشتند. برخی از این نیازها عبارتند از ارتباط متقابل وب اپلیکیشن‌ها با دوربین یا لیست شماره‌های تماس گوشی که برطرف کردن آن هم به راحتی امکان پذیر نبود.

Cordova برای مقابله با این محدودیت، مجموعه‌ای از رابط‌های برنامه کاربردی را برای توسعه قابلیت‌های بومی device، مانند لیست مخاطبین، دوربین، تشخیص دهنده‌ی تغییر جهت گوشی (accelerometer) و مانند این موارد، در نظر گرفته است.

Cordova شامل یک سری کامپوننت به شرح زیر است:

سورس کدی برای هر Container و برنامه محلی برای هر یک از سکوه‌های موبایل که پشتیبانی می‌شوند. container، کدهای Html5 را بر روی دستگاه (Device) رندر می‌کند. (در مطالب بعدی در مورد این مطلب توضیح خواهم داد)

مجموعه‌ای از رابط‌های برنامه کاربردی که امکان دسترسی به قابلیت‌های بومی دستگاه را به برنامه‌ی وبی که درون آن در حال اجرا است، می‌دهند.

مجموعه‌ای از ابزارها برای مدیریت فرآیند ایجاد پروژه، مدیریت پلاگین‌ها، ساخت (با استفاده از SDKهای محلی) برنامه‌های محلی و تست برنامه بر روی دستگاه موبایل یا شبیه ساز.

برای ساخت یک برنامه‌ی Cordova، در واقع شما یک وب اپلیکیشن می‌سازید و آن را داخل Container محلی، بسته بندی می‌کنید. سپس تست کرده و بعد از دیباگ می‌توانید اپلیکیشن را توزیع کنید.

فرآیند بسته بندی :

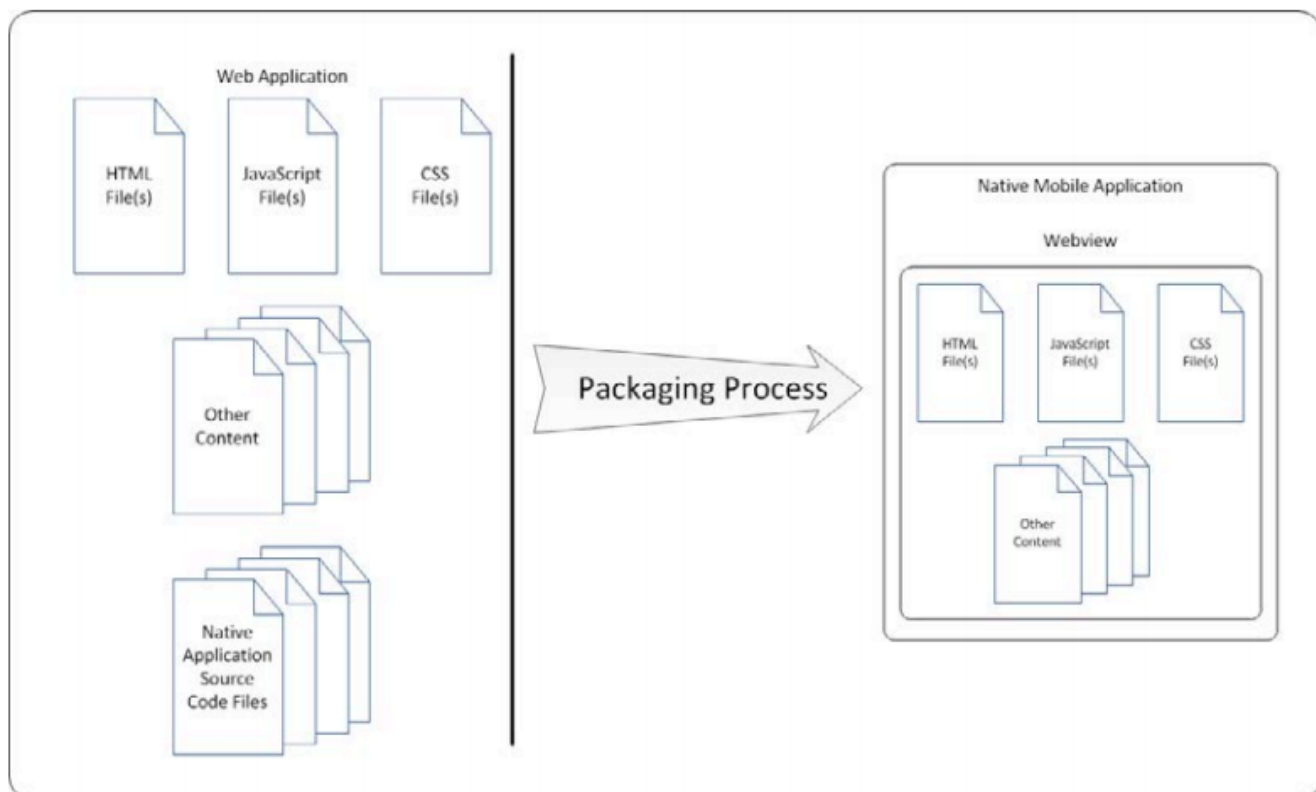


Figure 1.1 Apache Cordova Application Packaging Process

داخل اپلیکیشن محلی، رابط کاربری اپلیکیشن شامل یک صفحه‌ی نمایش که خود آن چیزی نیست به غیر از یک Web View که از فضای نمایش دستگاه استفاده می‌کند. زمانی که برنامه آغاز به کار می‌کند، برنامه‌ی وب نوشته شده، درون این web view لود میشود و کنترل‌های موجود، برای تعامل کاربر با برنامه‌ی وب، در اختیار آن قرار می‌گیرند. مانند تعامل کاربر با محتوا، در برنامه‌ها ی تحت وب، لینک‌ها، کدهای نوشته شده‌ی JS در فایل‌ها و یا حتی می‌تواند به اینترنت دسترسی داشته باشد و محتوا را از یک وب سرور تغذیه کند.

درباره Web Views

Web View جزء برنامه‌های بومی است که برای رندر کردن محتوای وب (به عنوان نمونه صفحه HTML) درون اپلیکیشن بومی یا صفحه نمایش استفاده می‌شود. در اصل Web View یک Wrapper برنامه نویسی شده قابل دسترس برای نمایش محتوای صفحات وب توکار است.

به عنوان مثال:

در اندروید با استفاده از WebView موجود در (Using android.webkit.WebView), در iOS با UIWebView موجود در (Using System/Library/Frameworks/UIKit.framework) به این هدف دست پیدا می‌کنند. وب اپلیکیشن ما درون این Container مانند سایر وب اپلیکیشن‌هایی است که هر روز با آنها سرو کار دارید و آنها را در مرورگر موبایل خود اجرا می‌کنید و می‌توانید بین صفحات Navigation داشته باشید. وب اپلیکیشن‌های معمول باید روی یک سرور هاست شوند. در برنامه نویسی چند سکویی با Cordova، این کار می‌تواند درون Cordova Application انجام گیرد.

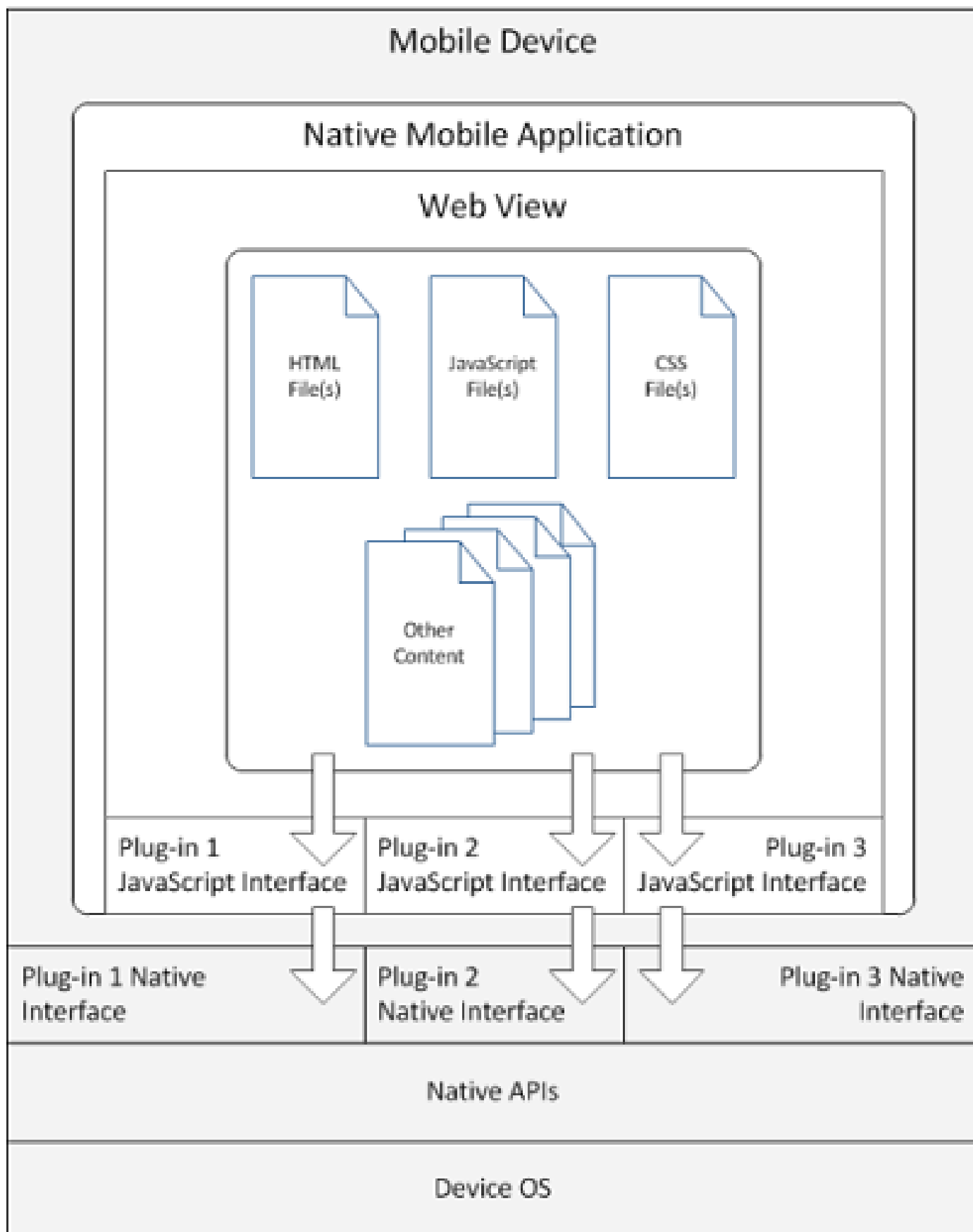
شاید سؤالی در ذهن شما وجود داشته باشد که مرورگر معمولاً به اپلیکیشن‌های موجود در دستگاه، سخت افزار و یا API‌های بومی دستگاه، دسترسی ندارد. برای مثال شاید بگویید که یک وب اپلیکیشن معمولاً به لیست مخاطبین با دوربین دستگاه و ... دسترسی ندارد.

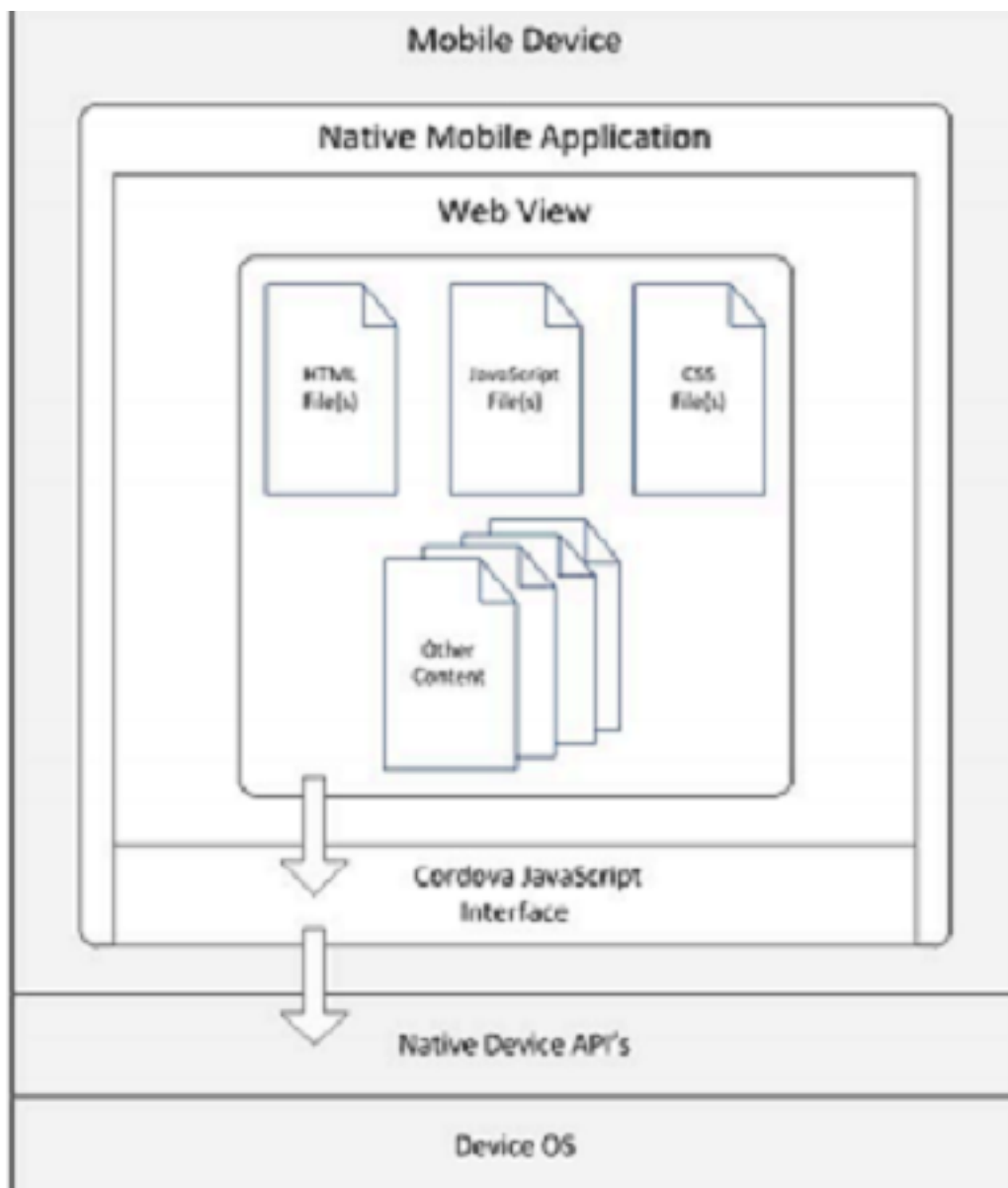
جواب : در واقع امکان دسترسی به این قابلیت‌ها توسط اپلیکیشن بومی (native mobile application) ایجاد می‌شود. Cordova مجموعه ای از API‌های جاوااسکریپت را به عنوان اهرم اجازه برای دسترسی برنامه وب درون cordova container به قابلیت‌های بومی دستگاه، در اختیار توسعه دهندگان قرار داده است.

این API ها در دو بخش پیاده سازی می شوند:

1- کتابخانه‌ی جاوااسکریپت که اجازه‌ی استفاده از قابلیت‌های بومی را به وب اپلیکیشن می‌دهد و کد بومی مشابه در Container اجرا می‌شود که مربوط است به بخش بومی این API ها. در اصل یک کتابخانه‌ی جاوا اسکریپت وجود دارد، اما بخش بومی API ها وابسته به سکوی (platform) انتخاب شده پیاده سازی می‌شود.

اگر شما از API های موجود استفاده نکنید، می‌توانید آنها را از کتابخانه جاوااسکریپت و native container حذف کنید. این کار به صورت دستی شاید خوشایند نباشد ولی چون در Cordova 3.0 همه‌ی API ها از بیرون وارد می‌شوند، می‌توانید با استفاده از بحث مدیریت پلاگین آن، پلاگین‌ها را اضافه یا حذف کنید. در بخش‌های بعد با مثال‌هایی عملی این مباحث را کار خواهیم کرد. **ادامه دارد...**





نظرات خوانندگان

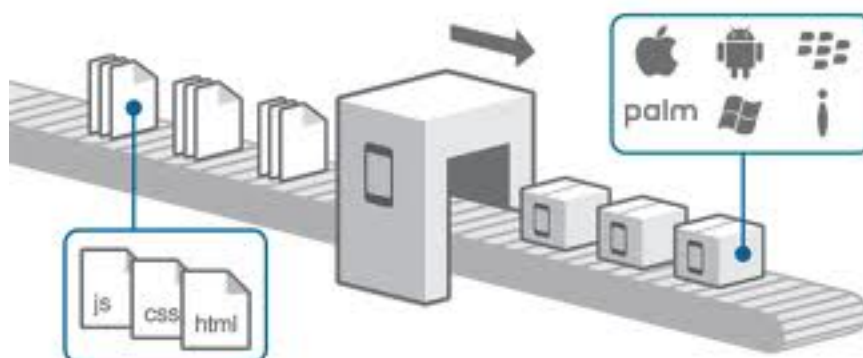
نویسنده: افشین عباسپور
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۲:۰۰

- خیلی ساده و روان توضیح دادید . متشکرم .
- 1- در مورد محدودیت‌ها هم توضیح بدید لطفا ...
 - 2- اینکه Performance این برنامه‌ها چطور است ؟
 - 3- امنیت برنامه‌های تولید شده چگونه است ؟ آیا سورس برنامه رو میتوان غیر قابل دسترس کرد ؟
- باز هم تشکر از این آموزش . منتظر آموزشهای بعدی و تکمیل این بحث هستم .

نویسنده: غلامرضا ربال
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۲:۴۵

حتما مقاله ای را برای این در ادامه تهیه خواهیم کرد تا به طور کامل این مباحث رو پوشش دهد. فعلا شاید [این](#) بتواند کمک کند. در ضمن سرعت این برنامه‌های نوشته شده با Cordova نسبت به برنامه‌های بومی اندکی کم است (مزایا و معایب خود را دارد) و برای اینکه سورس در دسترس نباشد روش هایی برای آن در نظر گرفته شده. [این مقاله هم مفید است.](#)

در [قسمت اول](#) ، با Apache Cordova آشنا شدیم. در این قسمت قصد داریم در مورد Phonegap، معایب و مزایای Cordova و روش نصب و راه اندازی آن را بر روی Visual Studio، خدمت شما ارائه دهیم.



توضیح مختصری در مورد Adobe Phonegap

در حوالی سال 2009 ، phonegap به واسطه‌ی استارت آپ بنام Nitobi با هدف ایجاد یک راه حل سورس باز برای ساخت اپلیکیشن‌های بومی موبایل با تکنولوژی‌های تحت وب، تولید شد. شرکت Adobe در حوالی سال 2011 ، Notobi را به همراه حق مالکیت phonegap خریداری کرد و هسته‌ی سورس باز آن را با نام Cordova به شرکت Apache اهدا کرد. نسبت بین Cordova و phonegap مانند نسبت بین مرورگر [Blink](#) و کروم است. در واقع phonegap ترکیبی است از Cordova و یک سری امکانات اضافی شرکت Adobe. تفاوت اصلی بین Cordova و Phonegap مربوط است به ابزارهای Command-Line و سرویس Build فون گپ است که در مقالات بعدی به آنها خواهیم پرداخت.

بیشتر : [اینجا](#) و [اینجا](#)

مزایای استفاده از Cordova:

محیط برنامه نویسی قدرتمند

هسته اصلی کدهای همه اپلیکیشن‌ها تولید شده شبیه به هم است

نیازی به یادگیری زبان‌های مربوط به هر پلتفرم را ندارید

کم هزینه و زمان کمتر

طراحی رابط گرافیکی سریع و منعطف به کمک CSS3 ، HTML5

برنامه نویسی آسان و سریع با Typescript ، javascript

قابلیت اجرا بر روی چندین پلتفرم مختلف (Android,iOS,Widnows Phone)

قابلیت استفاده از فریم‌ورک‌های تحت وب مانند Angular JS ، Bootstrap ، ...

قابلیت طراحی پلاگین برای ارتباط با سیستم عامل

مناسب برای برای برنامه‌های چت و استفاد از وب سرویس‌ها

مناسب برای ساخت بازیهای آنلاین و آفلاین با تکنولوژی‌های تحت وب

راحتی کار با آن برای برنامه نویسان تحت وب

[مخالفین و موافقین Cordova /Phonegap](#)

معایب استفاده از Cordova :

نداشتن ابزار گزارش خطاهای مناسب؛ در نتیجه برطرف کردن خطاها خسته کننده خواهد بود .

UI, UX اپلیکیشن‌ها باید به نحوی باشد که کاربر حس کند با نرم‌افزارهای بومی گوشی کار می‌کند.

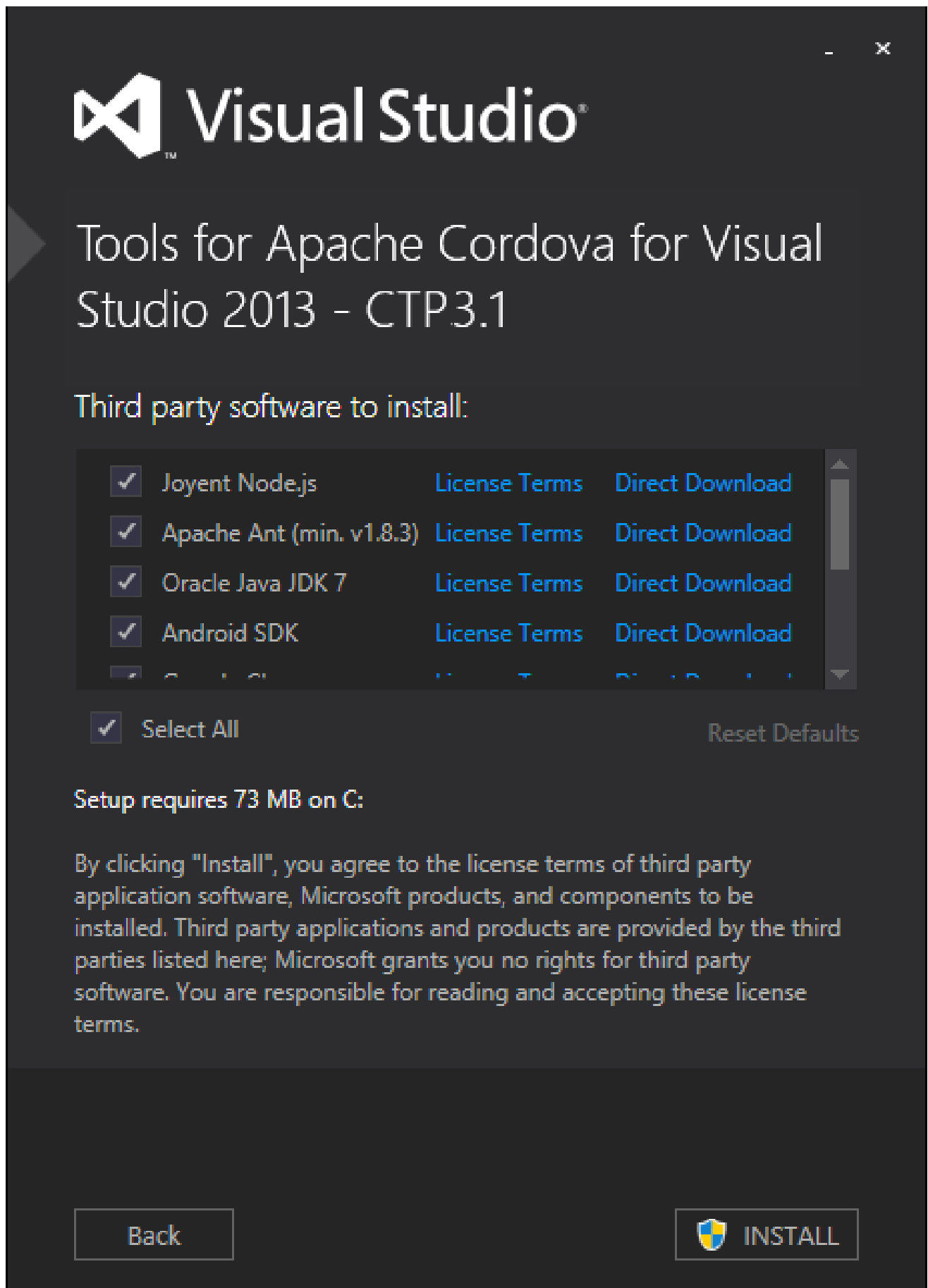
کاهش سرعت اجرایی جزئی نسبت به سایر برنامه‌ها (به دلیل استفاده از WebView)

عدم دسترسی مستقیم به سیستم عامل و امکانات آن

[اطلاعات بیشتر](#)

نصب اتوماتیک وابستگی‌ها

ابزارهایی که ما نیاز داریم:



لازم است تا Visual Studio 2013، با حداقل آپدیت 2 بر روی سیستم شما نصب باشد.

دانلود کنید: [Visual Studio Tools for Apache Cordova CTP3.1](#)

بعد از اتمام دانلود فایل، اقدام به نصب آن نمایید. در این حین، یک سری وابستگی‌های مربوط به خود را دانلود و نصب خواهد کرد. لیست وابستگی‌ها:

Node.js

Git CLI

Google Chrome

Apache Ant

Oracle Java JDK 7 (حتما نسخه x86 نصب شود)

Android SDK

SQLite For Windows Runtime

Apple iTunes

فایل نصاب، همه‌ی این وابستگی‌ها را به‌غیر از Android SDK، نصب می‌کند. [نسخه 32 بیتی](#) و [نسخه 64 بیتی](#) در آخر هم سیستم خود را راستارت کنید.
نصب دستی وابستگی‌ها:

اگر به هر دلیلی در نصب خودکار این وابستگی‌ها توسط نصاب با مشکل برخورد کردید، می‌توانید تک‌تک آنها را دانلود کرده و نصب کنید. لینک‌های مورد نظر را هم به همین دلیل قرار دادم.

node.js را از لینک مقابل دانلود کنید: [اینجا](#) (پیشنهاد می‌کنیم نسخه‌ی x86 آن را نصب کنید)

Google Chrome را نصب کنید

Git Command Line Tools را نصب کنید و توجه کنید که در هنگام نصب، گزینه مربوط به افزودن Git را به مسیر Command Prompt شما، انتخاب کرده باشید.

Apchage Ant را [دانلود](#) و در مسیری از سیستم خودتان قرار دهید.

[Java JDK 7 x86](#) را از لینک مشخص شده دانلود کنید و سپس عملیات نصب را انجام دهید.

Android SDK را از آدرس مشخص شده دانلود کنید. پکیج‌های مورد نیاز، به این SDK افزوده شده است. بعد از دانلود آن را در مسیری از سیستم خود قرار دهید.

Apple iTunes و SQLite را دانلود و نصب کنید.

اگر از ویندوز 7 استفاده می‌کنید ، WebSocket4Net را از لینک مقابل دانلود کنید ([اینجا](#)) و سپس فایل net45\Release\WebSocket4Net.dll در مسیر زیر کپی کنید:

```
ProgramFiles(x86)%\Microsoft Visual Studio%  
12.0\Common7\IDE\CommonExtensions\Microsoft\WebClient\Diagnostics\ToolWindows
```

ویژوال استودیو پیکربندی‌های مربوط به نرم افزارهای thrid-party (سوم شخص/ثالث: نرم افزارهایی که برای دستکاری بر روی سیستم عامل، توسط شرکت‌هایی غیر از شرکت‌های تولید کننده سیستم عامل تولید می‌شوند) را که شما نصب کرده‌اید، تشخیص می‌دهد و مسیرهای نصب آنها را درون متغیرهای محیطی (environment variables) به شکل زیر نگه می‌دارد:

ADT_HOME: به مسیر نصب اندروید اشاره می‌کند

ANT_HOME: به فولدری که Apache Ant در آن قرار دارد اشاره می‌کند

JAVA_HOME: به مسیر نصب جاوا اشاره می‌کند

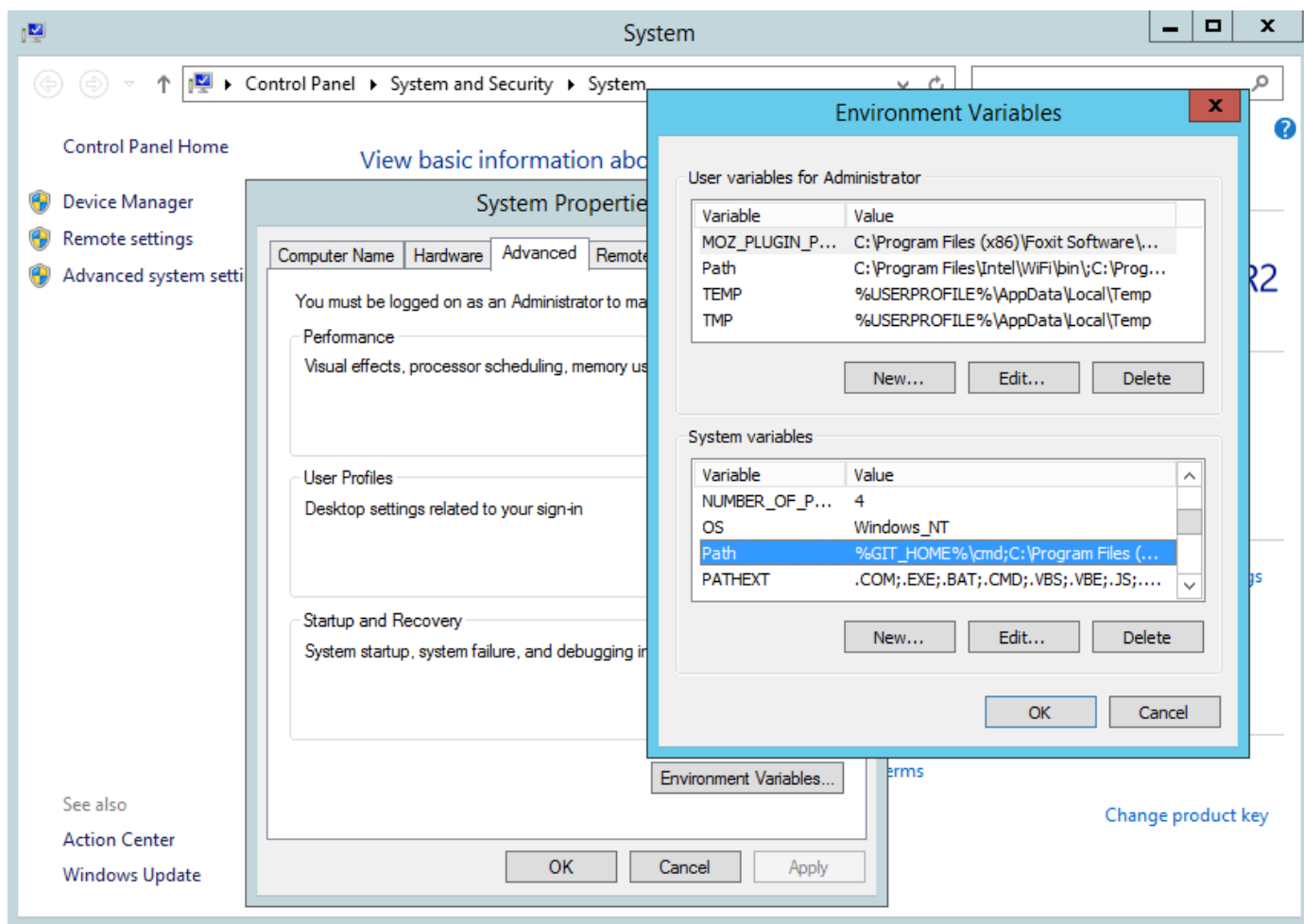
GIT_HOME: به مسیر نصب GIT اشاره می‌کند.

دقت کنید باید نام‌های متغیرها، دقیقاً به همین نام‌ها باشند.

برای تنظیم این متغیرها، به مسیر Control Panel\System and Security\System Advanced System Setting و گزینه‌ی Environment Variables را انتخاب کنید و در قسمت system variables، این 4 متغیری که ذکر شد را ایجاد کنید. سپس نیاز است این مسیرها را به system path اضافه کنید. برای این کار از همان قسمت system variables متغیر path را انتخاب کرده و گزینه‌ی ویرایش را بزنید و ابتدا محتویات آن را در یک فایل notepad کپی کنید و مسیرهای زیر را به اول آن اضافه کنید :

```
%GIT_HOME%\cmd;C:\Program Files (x86)\nodejs\;%JAVA_HOME%\bin;%ANT_HOME%\bin;  
%ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools; C:\ProgramData\Oracle\Java\javapath;
```

دقت کنید که مسیرهای ذکر شده فقط یکبار در کل فایل وجود داشته باشند و سپس کل محتوای فایل را کپی کرده و در قسمت path پیست کنید و با کلیک بر روی دکمه‌های OK کار را به اتمام رسانید.



نکته تکمیلی

نیازمندی Apache Cordova CTP3.1 : یکی از سیستم عامل‌های مقابل: Windows 7, Windows 8, Windows 8.1, or Windows Server 2012 R2. آپدیت 4 مربوط به ویژوال استودیو (دقت کنید قبل از نصب آپدیت 4 ویژوال استودیو باید نسخه قبلی Cordova CTP را حذف کنید(uninstall)) امکان توسعه اپلیکیشن‌های windows phone , windows برای کاربران ویندوز 7 وجود ندارد .

در مقاله‌ی بعدی یک پروژه جدید خواهیم ساخت .
منبع مفید برای نصب و راه اندازی : [اینجا](#) ادامه دارد...

نظرات خوانندگان

نویسنده: مرتضی ریسی
تاریخ: ۲۲:۲۲ ۱۳۹۴/۰۱/۰۶

سلام!

فایل [Visual Studio Tools for Apache Cordova CTP3.1](#) رو دانلود کردم. منتها موقع نصب میگه:

You must install Windows Management Framework 3.0...

چی هست؟ و چرا باید نصبش کنم؟ درضمن از آپدیت 4 ویژوال استادیو و ویندوز 7 استفاده می‌کنم. ممنون!

نویسنده: میثم ثوامری
تاریخ: ۲۳:۵۱ ۱۳۹۴/۰۱/۰۶

ممنون از مطلب آموزندتون
SDK و تمامی وابستگیها اندروید میتونید از [اینجا](#) دانلود کنید

نویسنده: غلامرضا ربال
تاریخ: ۰:۲ ۱۳۹۴/۰۱/۰۷

در واقع یک بروز رسان است که بعد از نصب این امکان رو به سیستم شما میدهد که به امکاناتی که در نسخه‌های جدید ویندوز اضافه شده است با دانلود آنها به آنها دسترسی داشته باشد. برای مثال PowerShell version 3 همراه ویندوز 8 منتشر شد ، با نصب این بروز رسان ، میتونید در ویندوز 7 هم به این امکانات دسترسی داشته باشید. [اطلاعات بیشتر](#)
از لینک زیر دانلود کنید [WMF 3.0](#)

نویسنده: مرتضی ریسی
تاریخ: ۰:۲۱ ۱۳۹۴/۰۱/۰۷

ممنون. به هر حال نصبش کردم. منتهای مراتب برای نصب وابستگی‌ها میگه نزدیک به 5 گیگ حافظه نیاز داره. توی تصویری که شما گذاشتین فقط 73 مگ نوشته. چطور دانلودش کنم؟ توی لیست وابستگی‌ها هم فقط WebSocket4Net اضافه شده که اونم حجمی نداره. به نظر شما مشکل کجاست؟

نویسنده: غلامرضا ربال
تاریخ: ۴:۹ ۱۳۹۴/۰۱/۰۷

طبق جستجویی که انجام دادم چیز مفیدی به دست نیاوردم و قبلا هم خودم با این مسئله برخورد نکردم شخصی هم در [وبلاگ Msdn](#) با همین مسئله ای روبرو شده بود که جوابی برای سوال او داده نشده. بنده حدس میزنم به خاطر امکاناتی هست که هم اکنون بر روی سیستم شما نصب نیست و همگام به این وابستگی‌ها نصب خواهد شد.
در کل پیشنهاد خود مایکروسافت هم ارتقا سیستم خود به ویندوز 8.1 است (برای بهرمندی از توسعه‌ی اپلیکیشن‌های ویندوز فون و ویندوز با استفاده از Cordova)

نویسنده: مرتضی ریسی
تاریخ: ۱۵:۵ ۱۳۹۴/۰۱/۰۷

سپاس. با همون 5 گیگ اقدام به نصب کردم. تعجب اینکه گویا همون 73 مگ رو نیاز داشت. (بر اساس زمان دانلود و نصب میگم)

نویسنده: آرش مصیر

تاریخ: ۱۷:۴۸ ۱۳۹۴/۰۱/۱۱

سلام و سپاس از مطلب خوبتون سوالی که مطرح میشه اینه که شنیده میشه تهیه اپلیکیشن با چنین framework هایی باعث میشه خروجی ایجاد شده حجیم تر و کندتر باشه. می‌خوام بدونم این نقصان چقدر حاد هست و این که آیا شما به شخصه اپلیکیشنی رو عملیاتی کردید و مثلا خروجیتون رو با خروجی Android Studio یا سایر ابزارها مقایسه کردید؟ آیا این framework کاملا مورد اطمینان هست؟

نویسنده: غلامرضا ربال
تاریخ: ۲۱:۵۷ ۱۳۹۴/۰۱/۱۱

سلام.

کندی این فریمورک مربوط میشود به طبیعت خود تکنولوژی که در آن استفاده شده است. استفاده از WebView به جای مرورگر هم باعث کاهش سرعت اجرای کدهای جاوا اسکریپت خواهد شد. ولی حاد هم نیست، الان اگر [این سایت](#) را مشاهده کنید، بیشتر برنامه‌های چت و بازی، با استفاده از این فریمورک ساخته میشود. در مقابل شما لازم نیست لزوما با زبان بومی پلتفرم مورد نظر خود درگیر باشید. میتوانید از فریمورک‌های JS , CSS که الآن هم روز به روز بهینه‌تر و محبوب‌تر میشود ، استفاده کنید. اگر کارایی این فریمورک برای ساخت بازی‌ها کافی باشد چرا باید برای ساخت اپلیکیشن‌های مورد نیاز ما بهینه نباشند. توجه کنید که این فریمورک ،وب اپلیکیشن ما را به زبان‌های بومی هر پلتفرم تبدیل نمی‌کند. زمانی که شما نیاز دارید به API های بومی سیستم عامل مورد نظری که دسترسی داشته باشید که برای آن امکان ساخت پلاگین وجود ندارد (در حال حاضر برای Calendar) در این صورت اگر برنامه شما نیاز به همچین امکانی دارد لذا نباید Cordova را انتخاب کنید.

بنده تا حالا سمت Android Studio نرفتم اصلا و صرفا به دلیل اینکه Cordova با فیلد من(وب) نزدیکی زیادی داشت سمت آن رفتم(یکی از مزایای آن)

در مقاله بعد در مورد این مبحث بیشتر تمرکز خواهیم کرد.

عنوان: شروع کار با Apache Cordova در ویژوال استودیو 3#

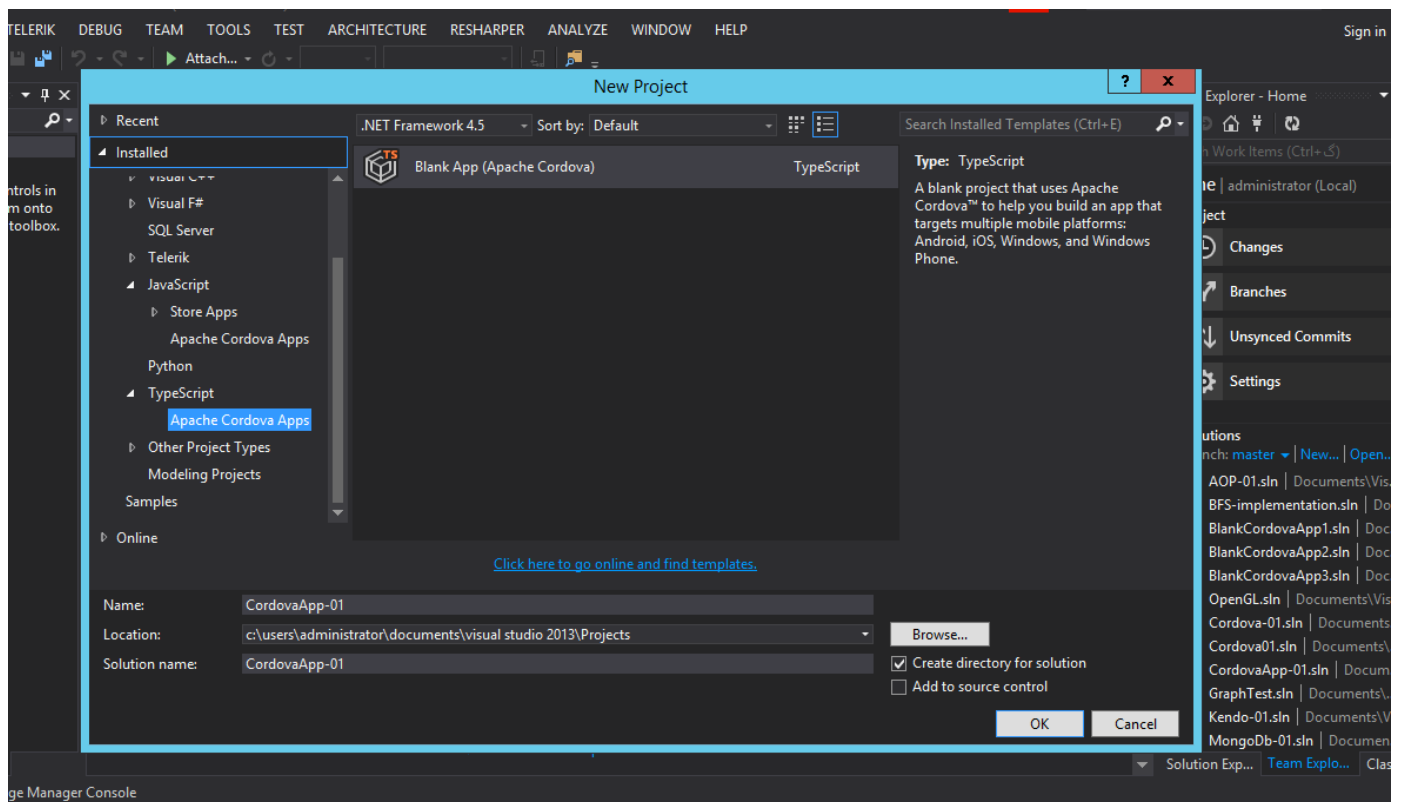
نویسنده: غلامرضا ربال

تاریخ: ۱۳۹۴/۰۱/۰۸

آدرس: www.dotnettips.info

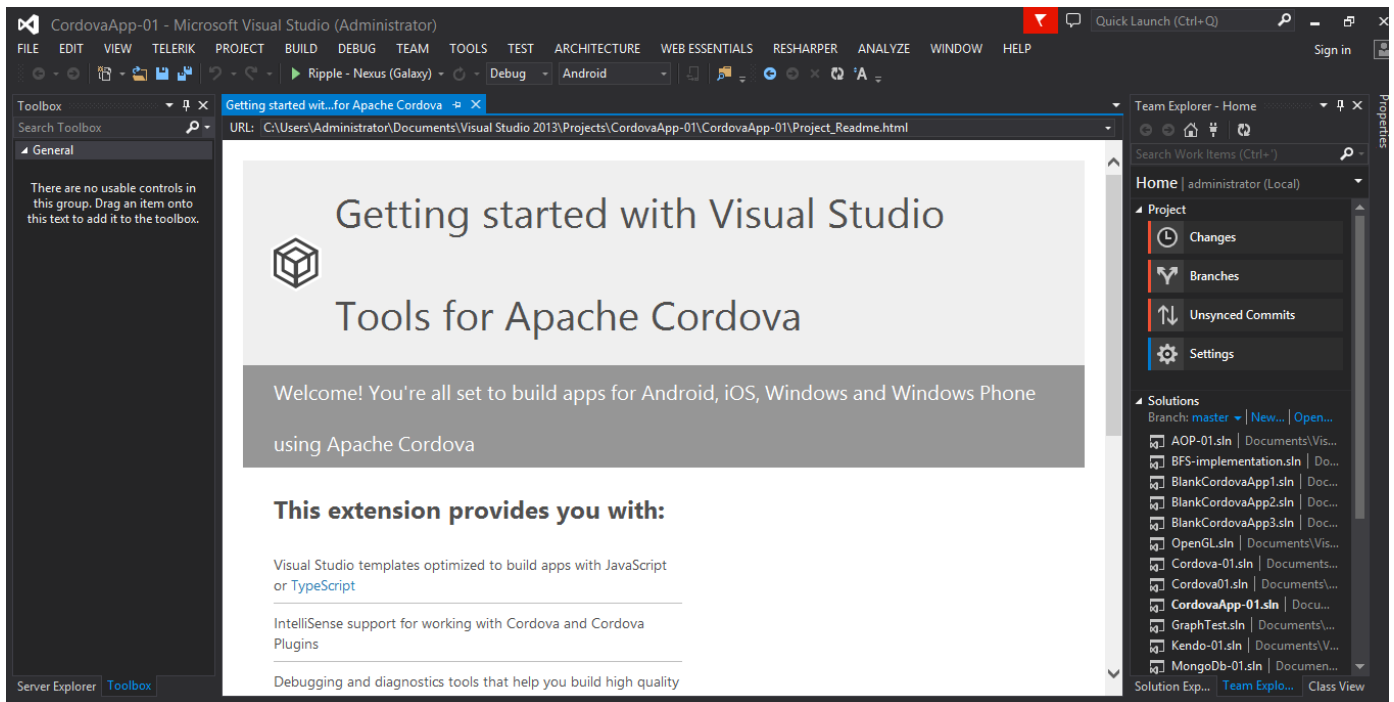
گروه‌ها: Visual Studio 2013, Apache Cordova, Mobile Programming

در قسمت قبل توانستیم ابزارهای لازم را برای Apache Cordova، نصب کنیم. در این قسمت یک پروژه‌ی ساده را ایجاد کرده و در مورد ساختار آن توضیح خواهیم داد. در ادامه‌ی مقالات از AngularJS، Bootstrap، TypeScript و jQuery Mobile هم در پروژه‌ها استفاده خواهیم کرد. برای شروع، از قسمت JavaScript یا TypeScript، یک پروژه‌ی از نوع Blank App ایجاد کنید. به شکل زیر:

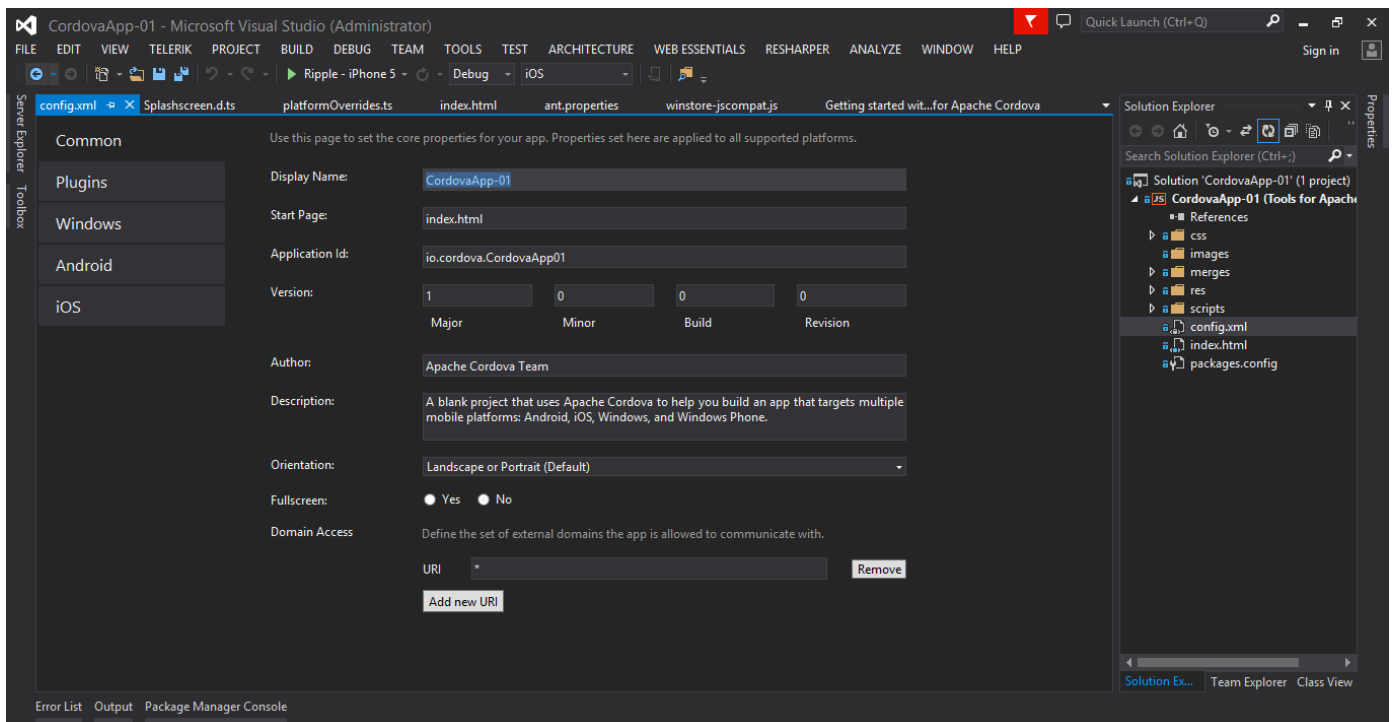


ترجیحا نوع TypeScript را انتخاب کردم. البته در داخل فایل ts، امکان نوشتن جاوا اسکریپت هم هست. بعد از ایجاد پروژه اگر با تصویری شبیه به تصویر زیر روبرو شدید، در نتیجه تنظیمات نصب و راه اندازی به درستی صورت گرفته است.

شروع کار با Apache Cordova در ویژوال استودیو 3#



اگر به قسمت solution explorer دقت کنید، فایلی به نام config.xml را مشاهده خواهید کرد. با کلیک بر روی این فایل، یک صفحه‌ی گرافیکی باز خواهد شد که این امکان را به شما می‌دهد که پلاگین‌های مورد نیاز خود، تنظیمات مربوط به نرم افزار تولیدی (مانند تنظیم ورژن ویندوزی که می‌خواهید app شما بر روی آن اجرا شود) و تنظیمات مربوط به هر یک از پلتفرم‌ها را به صورت مجزا در اختیار داشته باشید.

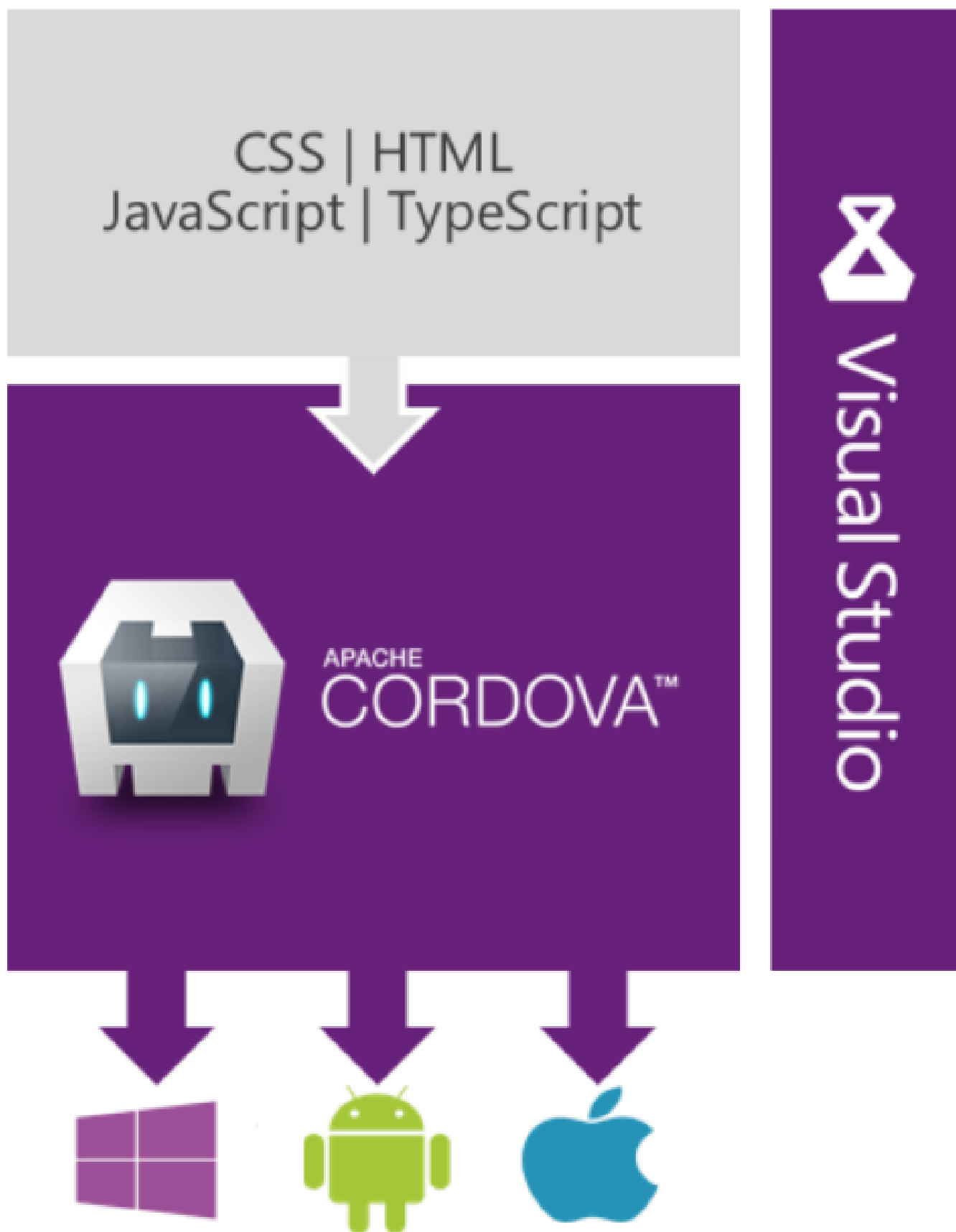


یک فایل index.html هم در قالب پیش‌فرض قرار داده شده که بعداً می‌توانید آن را تغییر دهید و یا صفحات دیگری را اضافه کنید.

همان طور که در قسمت‌های قبل گفته شد، قرار است ما یک وب اپلیکیشن طراحی کنیم و آن را درون Container بومی Cordova بسته بندی کنیم. لذا محدودیتی برای استفاده‌ی از کتابخانه‌های مرتبط با HTML ، CSS و JavaScript نداریم و در ادامه‌ی مقالات با مثال‌های متعددی از آن‌ها استفاده خواهیم کرد.

در فولدر `scripts-->typings-->cordova-->plugins` اینترفیس‌هایی که برای دسترسی به امکانات بومی دستگاه تلفن فعلا در Cordova پشتیبانی می‌شوند، قرار گرفته است.

برای استفاده از تکنولوژی‌های وب در محیط بومی دستگاه، در طی فرآیند کامپایل، Cordova یک اپلیکیشن را به وسیله دو چیز مهم که در زیر اشاره شده است، خواهد ساخت.



یک اپلیکیشن با یک کامپوننت WebView که با مرورگر یکپارچه شده است.

یه سری از منابعی که در داخل فایل‌های اپلیکیشن وب ما قرار دارند.

برای یکپارچه شدن API های Cordova با وب پیج موجود، اندکی کد نیاز داریم که برای انکار لینکی شبیه لینک زیر را در فایل html خود استفاده می‌کنیم که فقط بعد از کامپایل وجود خارجی دارد؛ به صورت زیر:

```
<script src="cordova.js"></script>
```

در پایان هم برای فهمیدن اینکه API های Cordova در دسترس هستند، می‌توانیم رخداد مربوط به deviceready را مدیریت کنیم؛ به صورت زیر:

```
document.addEventListener("deviceready", onDeviceReady, false);  
function onDeviceReady() { /* INIT */ }
```

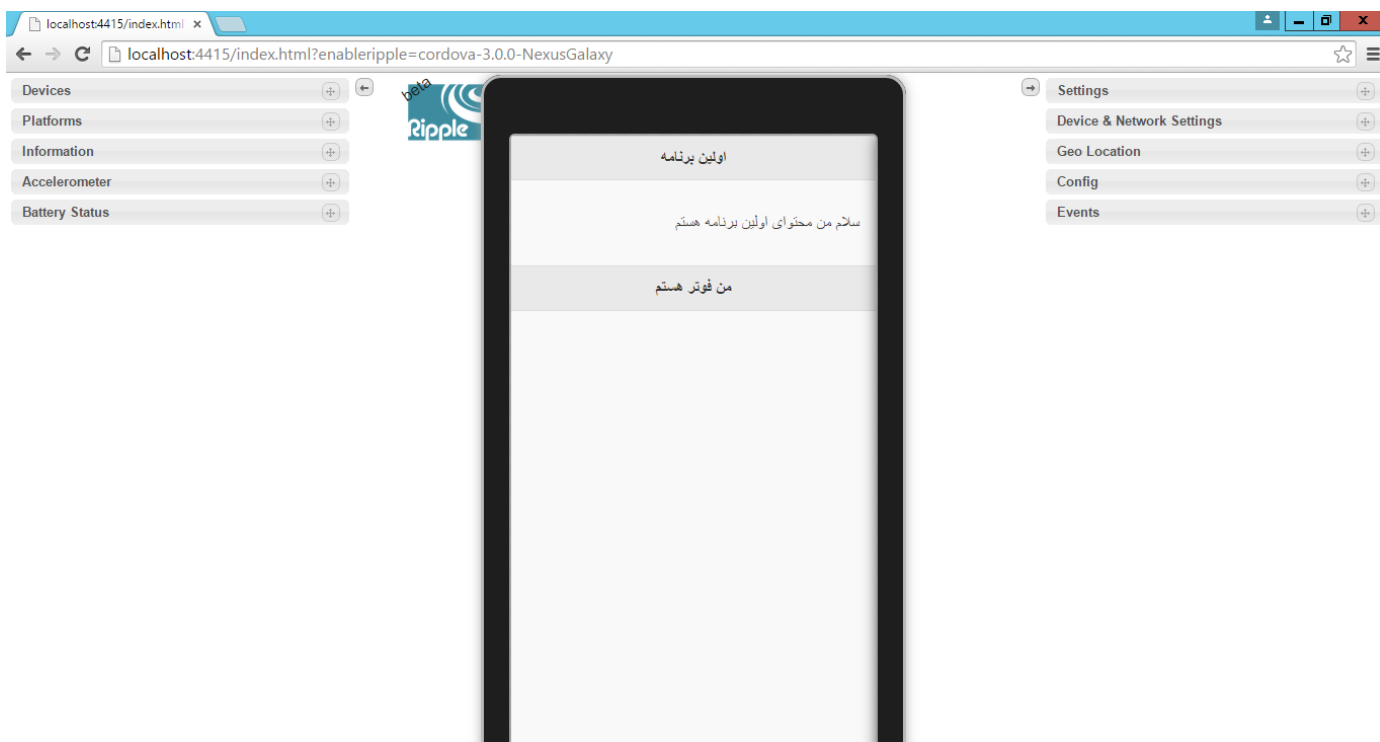
برای مدیریت رخدادهای مربوط به pause و resume هم که نشان دهنده‌ی ادامه برنامه (خارج شدن از حالت pause) و حالت تعلیق هستند، می‌توان به شکل زیر عمل کرد:

```
function onDeviceReady() {  
    // Handle the Cordova pause and resume events  
    document.addEventListener('pause', onPause, false);  
    document.addEventListener('resume', onResume, false);  
  
    // TODO: Cordova has been loaded. Perform any initialization that requires Cordova here.  
}  
  
function onPause() {  
    // TODO: This application has been suspended. Save application state here.  
}  
  
function onResume() {  
    // TODO: This application has been reactivated. Restore application state here.  
}
```

حال قصد داریم پروژه‌ی خود را که قرار است یک متن ساده را نشان دهد، با استفاده از شبیه ساز اجرا کنیم. برای این منظور از قسمت toolbar ویژوال استودیو، Solution Platform خود را انتخاب کنید و سپس می‌توانید شبیه ساز مورد نظر خود را انتخاب کرده و برنامه را اجرا کنید. در اینجا محیط مورد نظر من اندروید است و برای این منظور هم می‌توانم از شبیه ساز Android Emulator یا Ripple استفاده کنم. به دلیل سرعت کم شبیه ساز اندروید، می‌توانید شبیه ساز [YouWave](#) را دانلود و اجرا کرده و در قسمتی که شبیه ساز را از toolbar ویژوال انتخاب می‌کردید، این بار گزینه‌ی Device را انتخاب کنید. بعد از کامپایل برنامه‌ی شما، فایل apk تولید شده بر روی شبیه ساز نصب خواهد شد و شما قادر خواهید بود آنرا اجرا کنید.

نتیجه‌ی نهایی

با شبیه ساز Ripple



مطالعه بیشتر

[https://msdn.microsoft.com/en-us/library/dn879821\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/dn879821(v=vs.140).aspx)

<http://blog.falafel.com/getting-started-with-cordova-and-multi-device-hybrid-app-in-visual-studio>

<http://www.codeproject.com/Articles/860150/Visual-Studio-and-Apache-Cordova>

نکته : وقتی پروژه را برای اولین بار اجرا می‌کنید شاید کمی طول بکشد تا نتیجه‌ی نهایی را ببینید و آن هم به دلیل این است که ویژوال استودیو باید مجموعه‌ای از package های مورد نیاز Cordova را دانلود کند.

در مقاله بعد با jQuery Mobile آشنا خواهیم شد و یک مثال برای کار کردن با آن در نظر خواهیم گرفت.

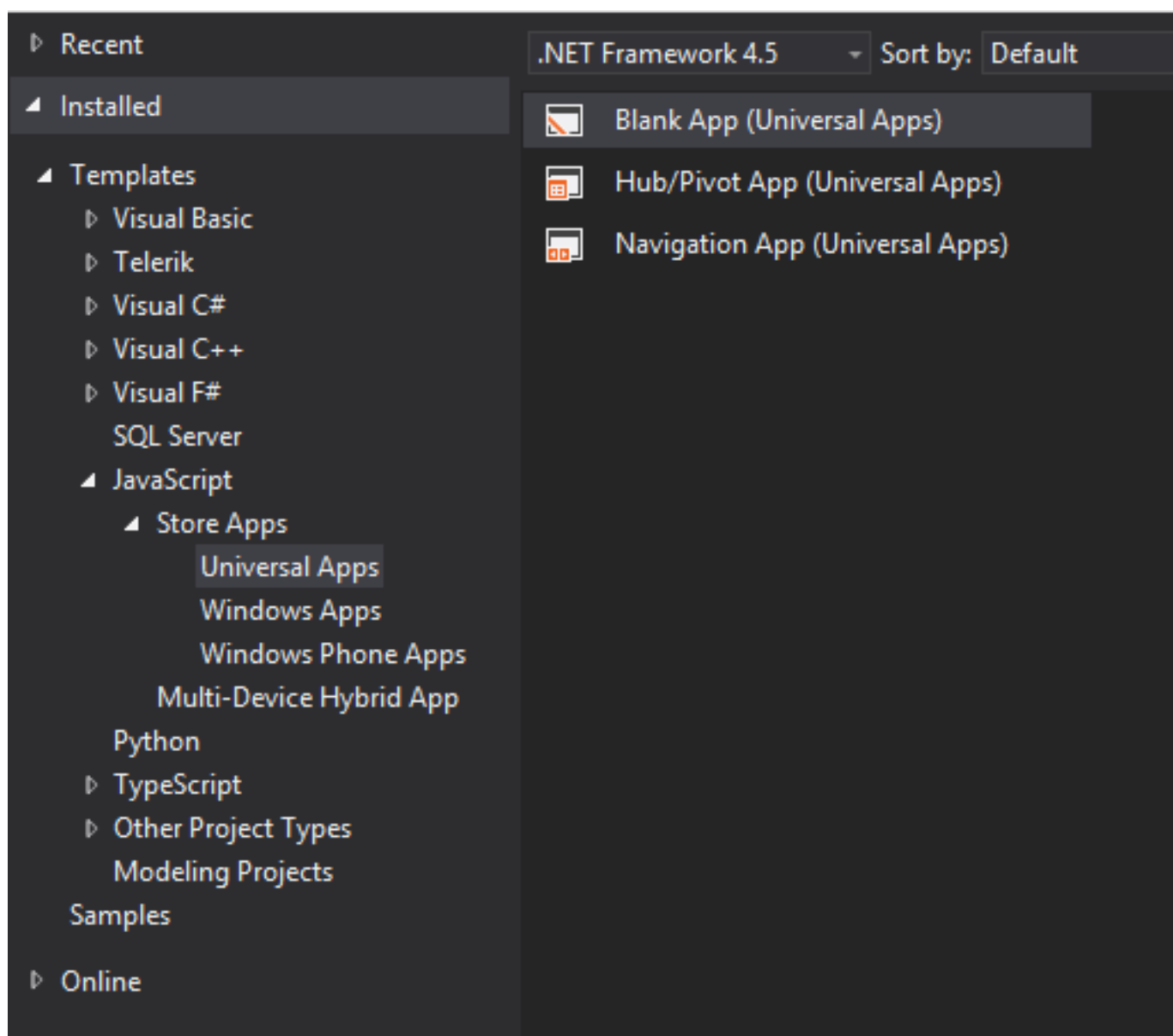
ادامه دارد ...

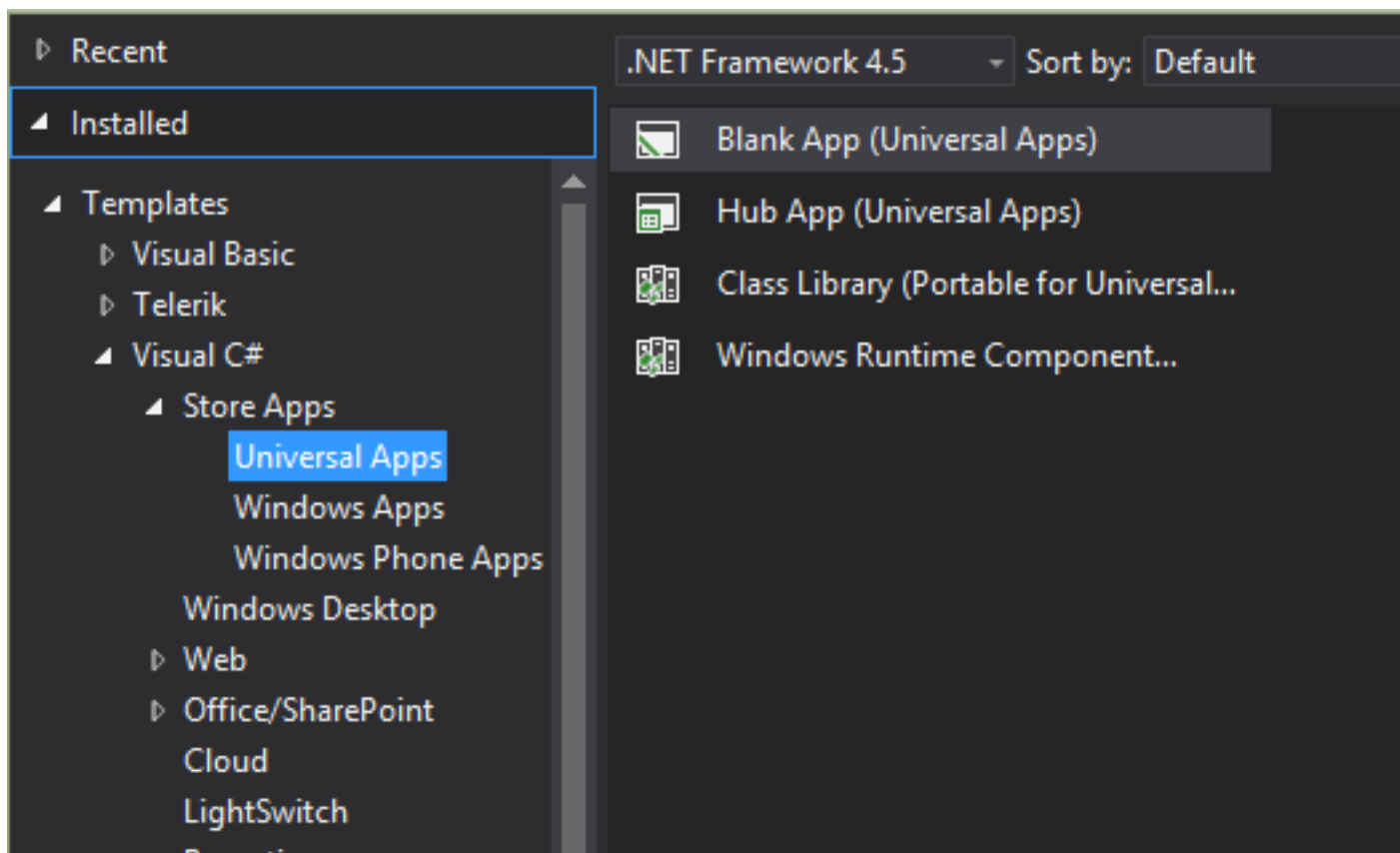
نظرات خوانندگان

نویسنده: خوزستان
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۰:۱۴

سلام ضمن تشکر بابت مقاله شما.
چند سوال :

- 1- آیا فقط با TypeScript و یا جاوا اسکریپت یا کتابخانه JQuery Mobile میتوان تمام نیازهای یک برنامه را تامین کرد ؟
- 2- دو روش دیگر برای ایجاد برنامه‌های موبایل وجود دارد :





تفاوت این دو روش ، با روش شما چیست ؟

3-در دو روش بالا زبانی مثل سی شارپ مورد استفاده قرار میگیرد ، در روش شما چطور ؟

4- آیا با توجه به محبوبیت زبان جاوا برای ساخت برنامه‌های اندرویدی ، روش مورد استفاده شما (cordova) میتواند با آن برابری کند ؟

تشکر

نویسنده:

محسن خان

تاریخ:

۱۰:۲۹ ۱۳۹۴/۰۱/۰۸

universal apps برای پلتفرم‌های مختلف مایکروسافت هست فقط. این مطلب یک قسمت اول هم داره: [شروع کار با Apache Cordova در ویژوال استودیو #1](#) . اونجا توضیح داده که این روش چند سکویی هست (یعنی فقط مختص به اندروید نیست). دسترسی به امکانات native دستگاه‌ها رو هم داره.

البته فقط این روش نیست که الان استفاده از جاوا اسکریپت رو شروع کرده برای توسعه‌ی برنامه‌های موبایل چندسکویی. شرکت تلریک هم اخیرا native script رو ارائه داده: <http://www.telerik.com/nativescript>

نویسنده:

غلامرضا ربال

تاریخ:

۱۱:۲۴ ۱۳۹۴/۰۱/۰۸

سلام.

1-بله البته در کنار html5 و css ، هدف اصلی Cordova هم همین است. در مقالات قبلی گفته شد که قرار است یک وب اپلیکیشن را در Cordova Container بسته بندی کنیم .

2- [این قسمت](#) و [اینجا](#) را مطالعه کنید . بنده کار نکردم . ولی فکر میکنم برای ساخت اپلیکیشن‌های همگانی ویندوز استفاده میشوند

- 3- خیر این از زیان سی شارپ استفاده نخواهد شد. هدف Cordova این نیست.
- 4- مطمئناً نمیتواند برابری کند حداقل در زمان حال ([مزایا و معایب خود را دارد](#)). اگر علاقه مند به توسعه پلاگین‌های Cordova باشید باز هم نیاز است از زبان بومی آن پلتفرم استفاده کنید.

نویسنده: خوزستان
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۲:۴۰

آیا از لحاظ کاربردی و قابلیت و توسعه ، بین [زمرین](#) و cordova تفاوتی هست ؟ آیا میکروسافتی که قصد خرید [زمرین](#) رو داشت الان بعنوان یک رقیب cordova رو معرفی کرده یا اینکه کارکرد متفاوتی دارند ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۲:۴۸

cordova در اصل محصول میکروسافت نیست. در قسمت دوم این بحث [شروع کار با Apache Cordova در ویژوال استودیو #2](#) بیشتر در این مورد بحث شده.

نویسنده: غلامرضا ربال
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۲:۵۴

[به نیاز شما بستگی دارد](#) .
[این مقاله](#) هم مفید خواهد بود و مقایسه خوبی انجام داده است.

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۴/۰۱/۰۹ ۱۶:۵۸

با تشکر از مطلب آموزنده شما Xamarin و Cordova رو همیشه با هم مقایسه کرد
Xamarin از زبان C# برای موبایل استفاده میکنه که البته زحمت کامپایل Mono بر عهده داره (Mono یک کامپایلر C# برای لینوکس و MonoToch برای IOS). و البته پولی
Cordova با استفاده از Javascript و Html5 اینکارو انجا میده

در [قسمت قبل](#) یک مثال ساده را کار کردیم. در این قسمت با jQuery Mobile آشنا شده و در پروژه‌ی خود استفاده خواهیم کرد.

توضیح تکمیلی در مورد ساختار فایل‌های پروژه

همان طور که در قسمت‌ها قبل گفته شد، تگ اسکریپت زیر

```
<script src="cordova.js"></script>
```

از استانداردهای Cordova است؛ وجود خارجی ندارد و بخشی از فرآیند ساخت برنامه است.

اگر توجه کنید فایلی با نام platformOverrides.js در فولدر scripts موجود در ریشه، خالی است اما در فولدر merges موجود در ریشه‌ی پروژه مربوط به هر پلتفرم و همان‌جا آن پلتفرم قرار دارد. برای مثال برای android، یک چنین دایرکتوری merges/android/scripts وجود دارد که درون آن فایلی به نام platformOverrides.js دیده می‌شود و اگر دقت کنید، همان‌جا فایلی موجود در فولدر scripts موجود در ریشه پروژه است که درون خود فایلی بنام android2.3-jscompat.js را فراخوانی می‌کند. (برای کمک به سازگاری کتابخانه‌های ثالث)

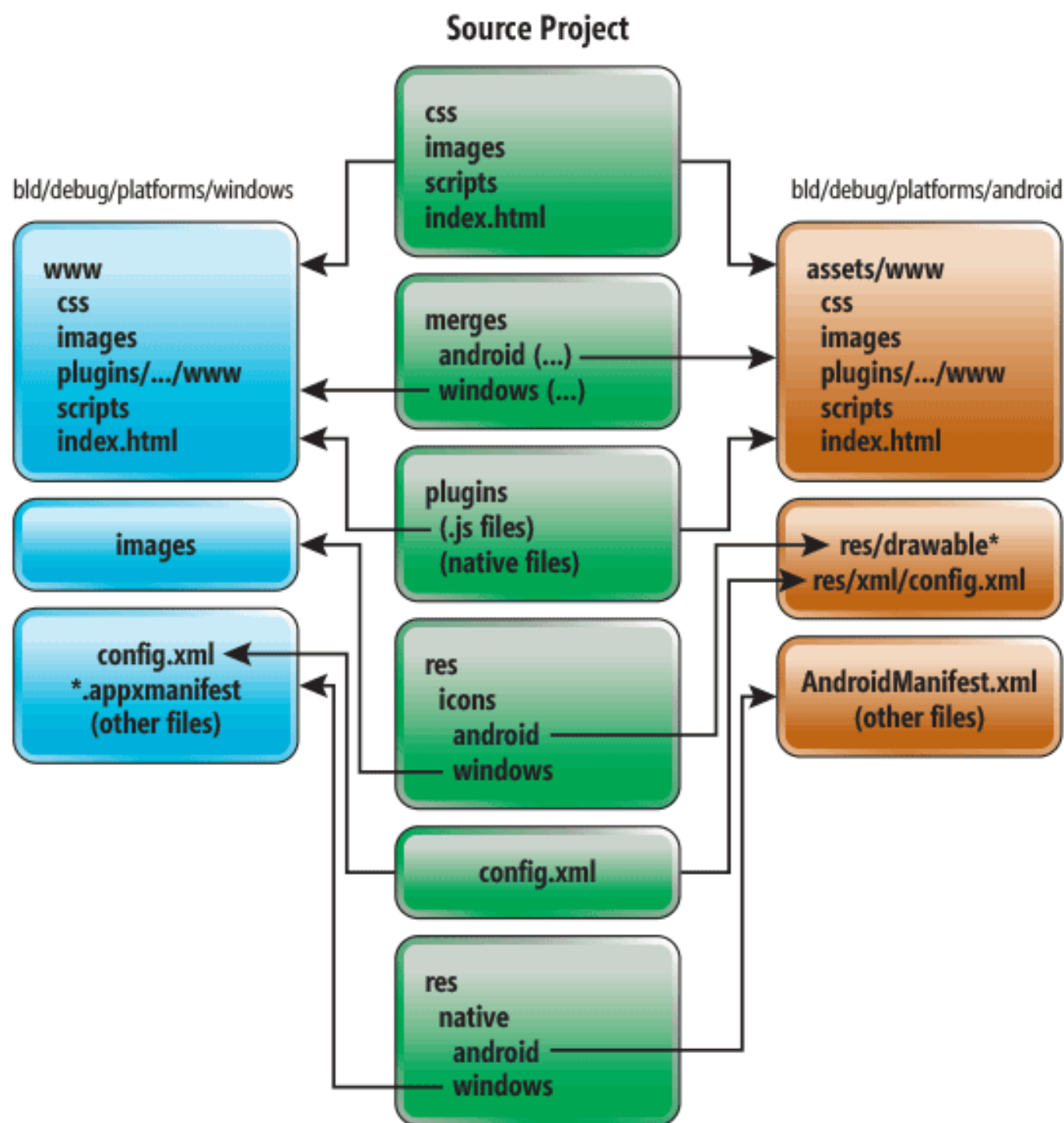
در زمان build، تمام فایل‌های موجود در "platformname" merges، در فولدرهای هم نامی در شاخه‌ی ریشه‌ی پروژه کپی شده و جایگزین فایل‌های قبلی خواهند شد.

مثال برای اندروید

در زمان ساخت (build) فایل scripts/platformOverrides.js با فایل merges/windows/scripts/platformoverrides.js جایگزین خواهد شد. این امکان برای فلد‌های css، images و بقیه‌ی آنها نیز امکان پذیر است. توجه داشته باشید این ادغام در سطح فایل‌ها و نه در سطح محتوای فایل‌ها انجام می‌شود.

نکته

برای محتوای موجود در فولدر res، قضیه فرق می‌کند. زیرا محتوای این resourceها برای اپلیکیشن پکیج ضروریست؛ پیش از آن که کدهای ما درون WebView یا host رندر شوند. باید توجه کرد که این فولدر به جهت اینکه منابع اصلی را (با توجه به پلتفرم باید از فایل‌های مشخص آن برای تشخیص ساختار فولدرهای اپلیکیشن پکیج استفاده کند) در بر دارد و این منابع باید در زمان ساخت پروژه تشخیص داده شوند.



رویدادهای بومی

در زیر تعدادی از رخدادهایی که در Cordova گنجانده شده‌اند تا اپلیکیشن ما از رخدادهای دستگاه با خبر شوند، نشان داده شده است. برای تست آنها به راحتی بعد از اجرای برنامه توسط شبیه ساز Ripple می‌توانید از قسمت Events، رخداد مورد نظر را شبیه سازی کنید:

```
(function () {
    "use strict";

    document.addEventListener( 'deviceready', onDeviceReady.bind( this ), false );

    function onDeviceReady() {
        // Handle the Cordova pause and resume events
        document.addEventListener( 'pause', onPause.bind( this ), false );
    }
})
```

```
document.addEventListener('resume', onResume.bind(this), false);
document.addEventListener('menubutton', onMenuButton.bind(this), false);
document.addEventListener('backbutton', onBackButton.bind(this), false);
//document.addEventListener('searchbutton', onResume.bind(this), false);
//document.addEventListener('endcallbutton', onResume.bind(this), false);
//document.addEventListener('offline', onResume.bind(this), false);
//document.addEventListener('online', onResume.bind(this), false);
//document.addEventListener('startcallbutton', onResume.bind(this), false);
//document.addEventListener('volumedownbutton', onResume.bind(this), false);
//document.addEventListener('volumeupbutton', onResume.bind(this), false);

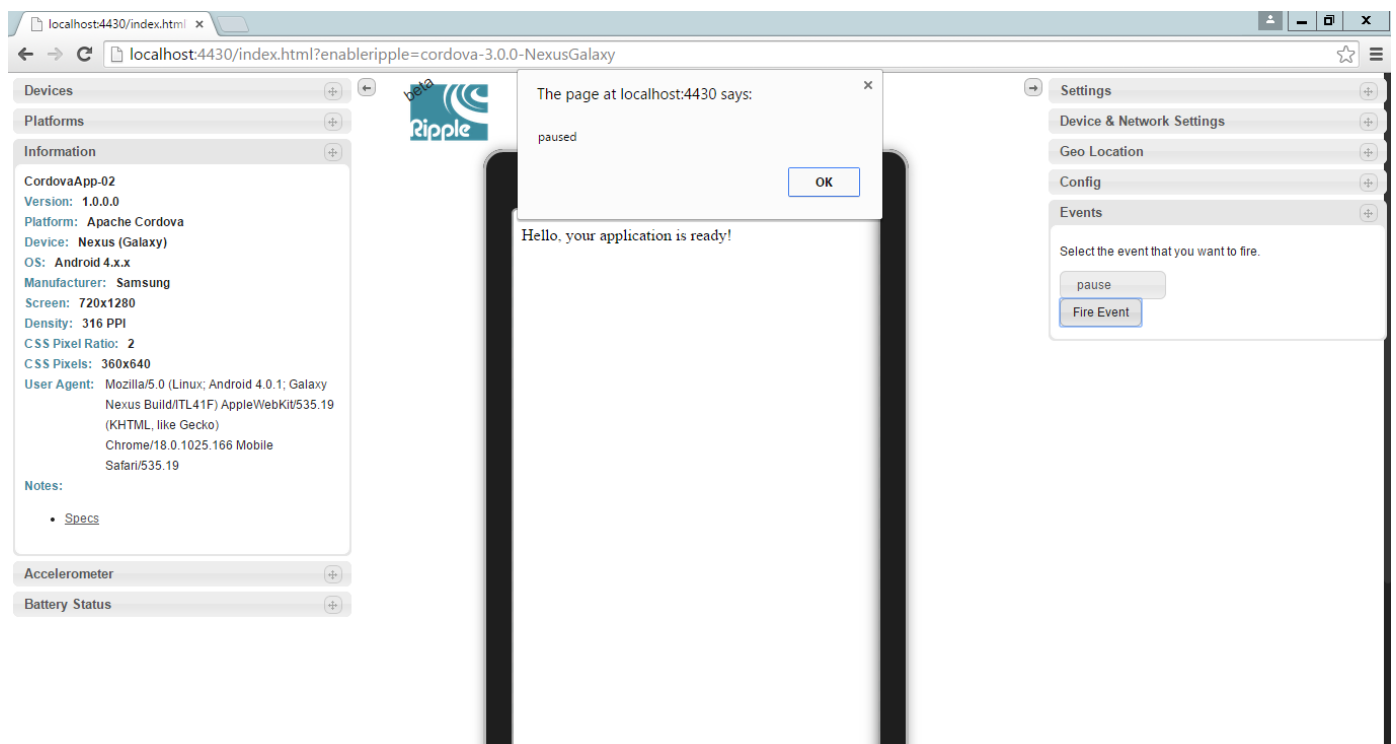
// TODO: Cordova has been loaded. Perform any initialization that requires Cordova here.
};

function onPause() {
    // TODO: This application has been suspended. Save application state here.
    alert("paused");
};

function onResume() {
    alert("resume");
};
function onMenuButton() {
    alert("menu");
};

function onBackButton() {
    alert("back button");
};

} )();
```



در مقالات آینده از افزونه‌های موجود، برای مدیریت رخدادهای باتری سیستم استفاده خواهیم کرد. **jQuery Mobile جی کوئری موبایل**، یک فریمورک (UI Framework) جدید با قابلیت استفاده‌ی آسان برای ساخت اپلیکیشن‌های چند سکویی موبایل است. با استفاده از این فریمورک شما قادر خواهید بود اپلیکیشن‌های موبایل بهینه شده برای اجرا بر روی تمام تلفن‌ها، دسکتاپ و تبلت‌ها را بسازید. علاوه بر این، جی کوئری موبایل می‌تواند یک فریمورک ایده آل برای توسعه دهندگان و طراحان وب که قصد ساخت اپلیکیشن‌های غنی وب برای موبایل را دارند، باشد.

Supported Devices

Phones/Tablets
 Android 1.6+
 BlackBerry 5+
 iOS 3+
 Windows Phone 7
 WebOS 1.4+
 (Symbian (Nokia S60
 Firefox Mobile Opera Mobile 11+
 Opera Mini 5+
 Desktop browsers
 Chrome 11+
 Firefox 3.6+
 Internet Explorer 7+
 Safari

برای نصب jQuery Mobile کافی است دستورات زیر را در package manager console ویژوال استودیو استفاده کنید:

```
PM>install-package jquery
```

```
PM>install-package jquery.mobile.rtl
```

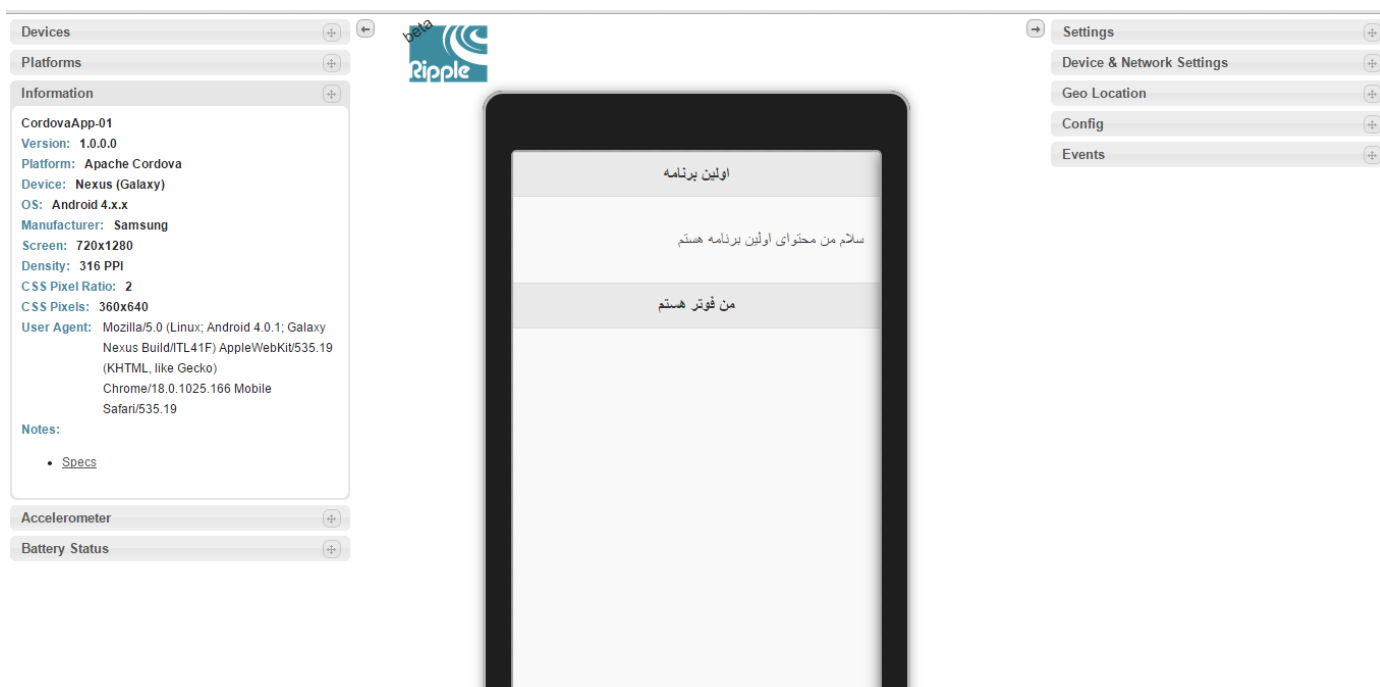
بعد از دانلود فایل‌های مورد نظر خود، فولدری بنام jquery.mobile.rtl در ریشه پروژه ایجاد خواهد شد. به ترتیب فایل‌های rtl.jquery.mobile-1.4.0.js و rtl.jquery.mobile-1.4.0.css موجود در زیر شاخه‌های فلدر مذکور را به head و آخر body فایل index.html اضافه کنید.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>CordovaApp01</title>

  <!-- CordovaApp01 references -->
  <link href="css/index.css" rel="stylesheet" />
  <link href="jquery.mobile.rtl/css/themes/default/rtl.jquery.mobile-1.4.0.css" rel="stylesheet" />
</head>
<body>
  <div data-role="page" id="page1">
    <div data-role="header">
      <h1>اولین برنامه</h1>
    </div>
    <div data-role="content">
      <p>سلام من محتوای اولین برنامه هستم</p>
    </div>
    <div data-role="footer">
      <h1>من فوتر هستم</h1>
    </div>
  </div>
  <!-- Cordova reference, this is added to your app when it's built. -->
  <script src="scripts/jquery-2.1.3.min.js"></script>
  <script src="cordova.js"></script>
  <script src="scripts/platformOverrides.js"></script>
  <script src="scripts/index.js"></script>

  <script src="jquery.mobile.rtl/js/rtl.jquery.mobile-1.4.0.js"></script>
</body>
</html>
```

در تکه کد بالا ما یکی از ویجت‌های jQuery Mobile را استفاده کردیم و با استفاده از ویژگی data-role که برای div اصلی با page مقدار دهی شده است، یک کانتینر (page container) برای ویجت page جی کوئری موبایل تعریف شده است. نتیجه‌ی نهایی به شکل زیر خواهد بود:



در مقاله‌ی بعد به استفاده از plugin‌ها خواهیم پرداخت.

ادامه دارد...

همانطور که در [قسمت قبل](#) گفته شد، در این قسمت با روش کار jQuery Mobile و pluginهای مربوط به Cordova آشنا خواهیم شد.

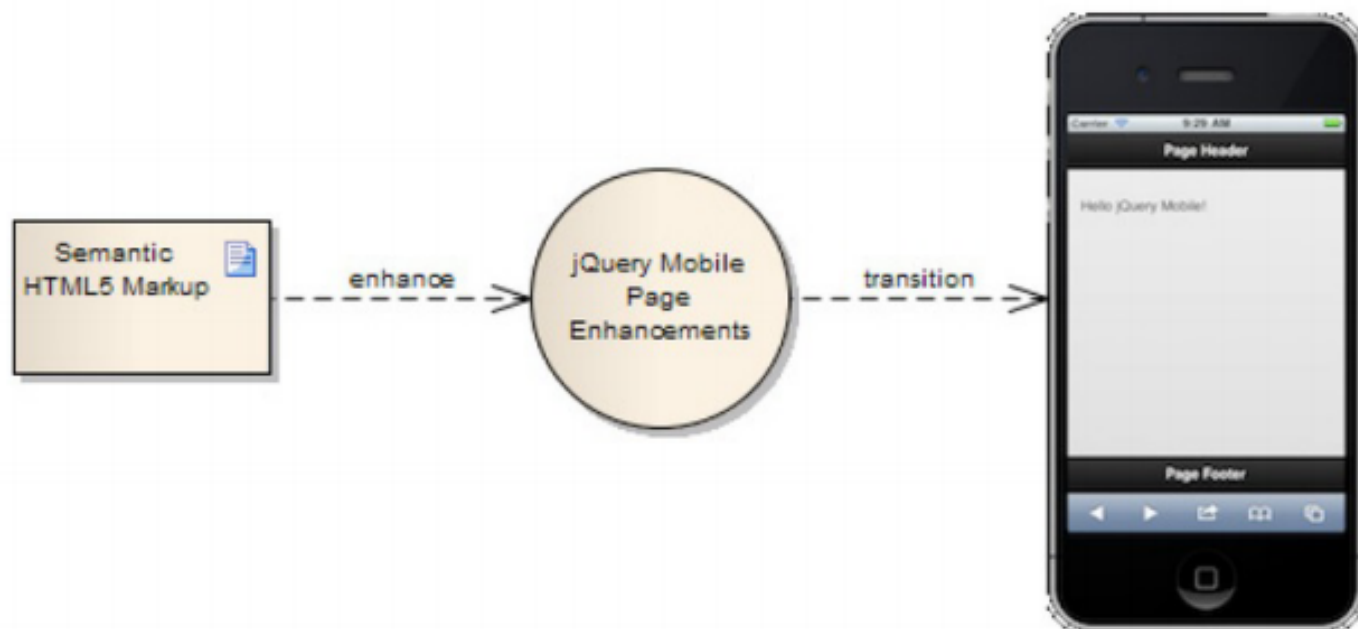
تگ متای زیر برای تنظیمات مربوط به viewport است و برای jQuery Mobile توصیه می‌شود.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Title</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
```

device-width نشان می‌دهد که می‌خواهیم مقیاس محتوای ما به اندازه‌ی عرض دستگاه (device) مورد نظر باشد و initial-scale هم مقدار زوم را برای Web page ما مشخص می‌کند. شما می‌توانید با مقدار دهی user-scalable=no هم امکان تغییر زوم را به کاربر ندهید. این متا تگ را در تمام صفحات html خود بعد از تگ title قرار دهید.

روال کار jQuery Mobile

برای اینکه بتواند سند HTML ما را برای استفاده‌ی در موبایل بهینه کند، ابتدا آن را لود می‌کند و سپس بر اجزایی که با ویژگی data-role علامت گذاری شده‌اند، CSS3 بهینه شده برای موبایل را اعمال می‌کند.



از آنجایی که مستندات jQuery Mobile به قدر کافی کامل هست، نیازی نیست تا در مورد تک تک آنها مثال بزنیم و از اصل مطلب دور شویم. در هر مثالی که زده خواهد شد، در صورت استفاده از ویژگی خاص، با آن آشنا خواهیم شد.

[لیست کامل اتریبیوت‌های data- به همراه مقادیری که می‌پذیرند](#)

[دموی مربوط به ویجت‌ها](#)

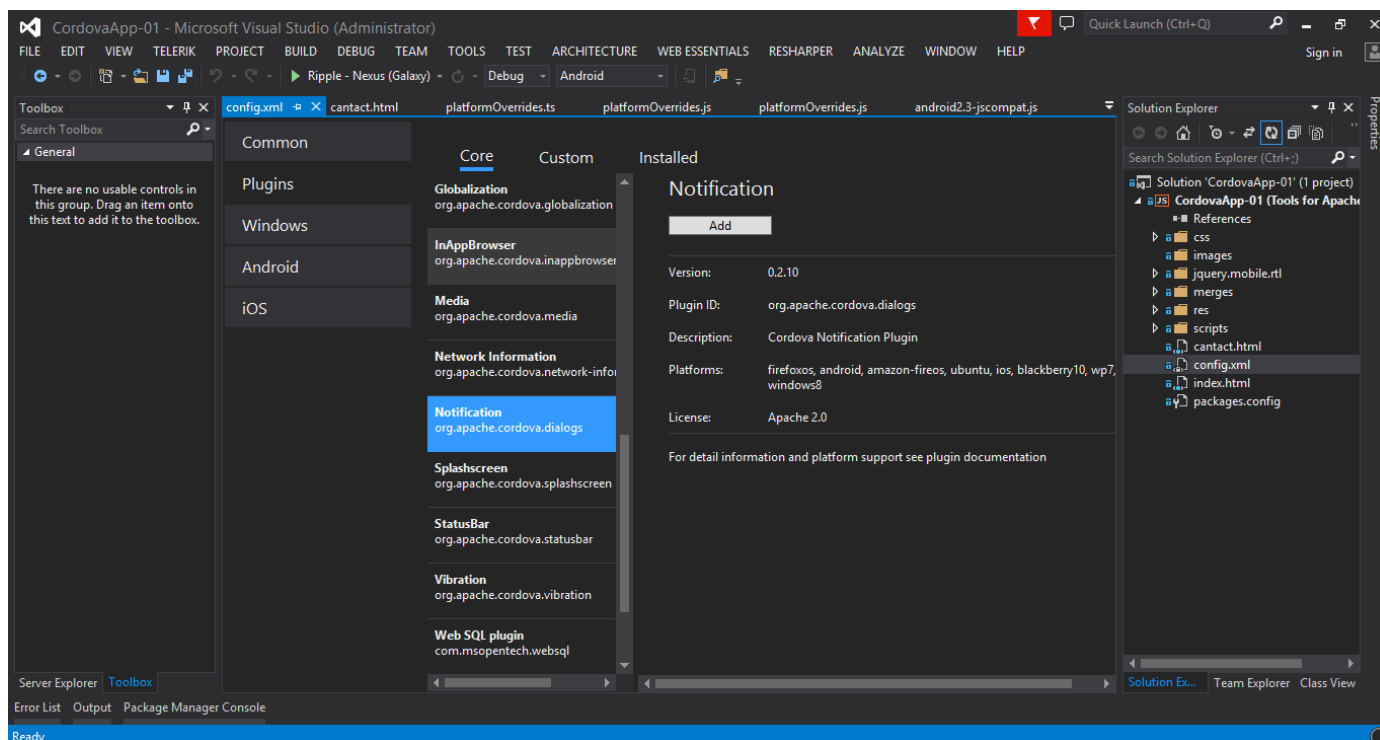
[لیست تمام رخدادهای](#)

شما می‌توانید از امکانات [Theme Roller](#) برای شخصی‌سازی تم‌های مورد نیاز استفاده کنید.

[لیست کامل کلاس‌های CSS](#)

Cordova Plugins

از این قسمت <http://plugins.cordova.io/#/viewAll> و این قسمت <http://plugreg.com/plugins> می‌توانید سراغ پلاگین‌های مورد نیاز خود بگردید. برای مثال وارد بخش کانفیگ پروژه شده و از قسمت plugins و تب Core یکسری از پلاگین‌هایی را که در Cordova گنجانده شده است، مشاهده می‌کنید. با کلیک بر روی دکمه‌ی Add می‌توانید آن را دانلود کرده و از API‌های آن استفاده کنید.



برای مثال پلاگین Notification را به پروژه اضافه می‌کنم. سپس یک فایل js را با نام custom.js به فولدر scripts در ریشه پروژه اضافه کرده و محتوای فایل‌های index.html, custom.js را به شکل زیر در نظر می‌گیرم:

```
$(function() {
    $("#alert").on('tap', function(event) {
        navigator.notification.alert("اطلاعات ذخیره شد", null, "alert", "تایید");
    });

    $("#prompt").on('tap', function(event) {
        navigator.notification.prompt("برای تأیید نام خود را وارد کنید", onPrompt, "prompt", "تایید", "نام خود را بنویسید");
    });
});
```

```

function onPrompt(results) {
    navigator.notification.alert(results.buttonIndex + "\n" + results.input1, null);
}
$("#confirm").on('tap', function(event) {
    navigator.notification.confirm("حذف انجام شود؟", onConfirm, "confirm", ["بله", "خیر"]);
});

function onConfirm(buttonIndex) {
    navigator.notification.alert(buttonIndex, null);
}
$("#beep").on('tap', function(event) {
    navigator.notification.beep(1);
});
});

```

رخداد tap زمانی صادر می‌شود که کاربر، دکمه‌ی مورد نظر را لمس کند و یکی از رخدادهای jQuery Mobile می‌باشد. بعد از نصب پلاگین Notification، با استفاده از navigator.notification می‌توانید به متدهای مورد نظر که در بالا مشخص است، دسترسی پیدا کنید.

برای آشنایی با این پلاگین می‌توانید [داکیومنت](#) آن را مطالعه کنید.

در کد بالا با استفاده از متدهای callback توانسته‌ایم اطلاعاتی در مورد نوع عملکرد کاربر با notification ما بدست آوریم.

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>CordovaApp01</title>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <!-- CordovaApp01 references -->
    <link href="css/index.css" rel="stylesheet" />
    <link href="jquery.mobile.rtl/css/themes/default/rtl.jquery.mobile-1.4.0.css" rel="stylesheet" />
</head>
<body>
<div data-role="page" id="page1">
    <div data-role="header">
        <h2>Notification تست پلاگین</h2>
    </div>
    <div data-role="content">
        <a href="#page2" data-transition="pop" data-rel="dialog" data-role="button" data-inline="true" data-icon="back">page 2</a>

        <button data-role="button" id="alert" data-inline="true" >alert</button>
        <button data-role="button" id="confirm" data-inline="true">confirm</button>
        <button data-role="button" id="beep" data-inline="true" >beep</button>
        <button data-role="button" id="prompt" data-inline="true" >prompt</button>

    </div>
    <div data-role="footer">
        <h2>من فوتر هستم</h2>
    </div>
</div>
<div data-role="page" id="page2">
    <div data-role="header">
        <h1>Header</h1>
    </div>
    <div data-role="content">
        Content
    </div>
    <div data-role="footer">
        <h1>Footer</h1>
    </div>
</div>
<!-- Cordova reference, this is added to your app when it's built. -->
<script src="scripts/jquery-2.1.3.min.js"></script>
<script src="cordova.js"></script>
<script src="scripts/platformOverrides.js"></script>

```

```
<script src="scripts/index.js"></script>
<script src="jquery.mobile.rtl/js/rtl.jquery.mobile-1.4.0.js"></script>
<script src="scripts/custom.js"></script>
</body>
</html>
```

در کد بالا 4 تا button دیده می‌شود که ویژگی data-role آنها مقدار button در نظر گرفته شده‌است تا توسط jQuery Mobile به عنوان button شناخته شوند و استایل‌های لازم بر روی آن‌ها اعمال گردد. قرار است طبق کد js ایی که نوشته‌ایم، با لمس کردن هر کدام از دکمه‌ها، notification هایی نمایش داده شوند.

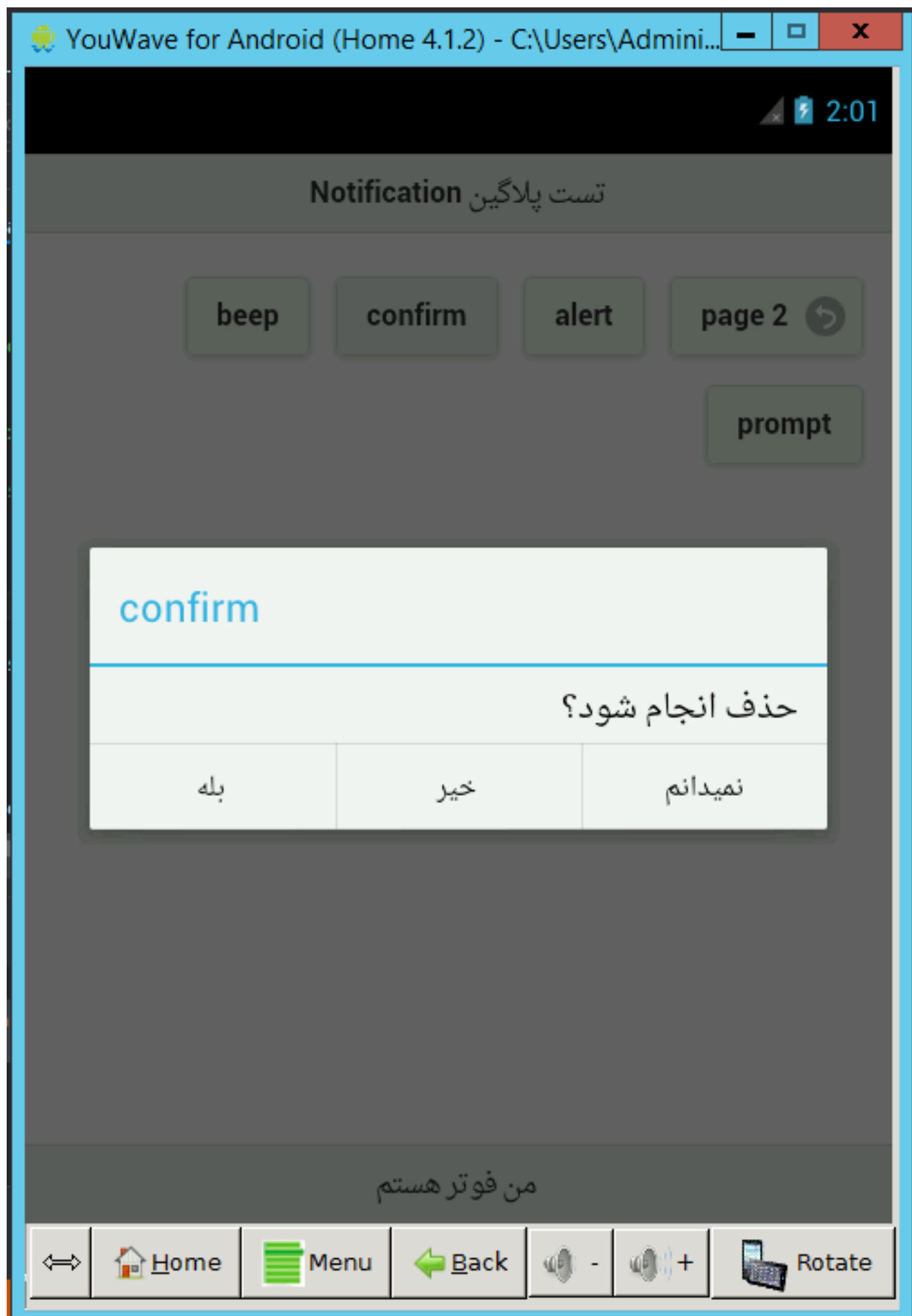
برای اینکار شبیه ساز [YouWave](#) را دانلود کرده و نصب کنید. سپس در قسمت toolbar ویژوال، گزینه‌ی Device را به جای شبیه ساز Ripple انتخاب کنید. نرم افزار youwave را اجرا کنید حال اگر برنامه را اجرا کنید با خطای زیر مواجه خواهید شد:

```
Error447C:\Users\Administrator\Documents\Visual Studio 2013\Projects\CordovaApp-01\CordovaApp-01\bin\Debug\platforms\android\cordova\node_modules\q\q.js:126CordovaApp-01
Error448throw e;CordovaApp-01
Error449^CordovaApp-01
Error450Error : DEP10201 : Failed to deploy to device, no devices found.CordovaApp-01
```

مشخصاً خطا، مبنی بر پیدا نشدن دستگاه خارجی است. برای رفع این مشکل می‌بایست شبیه ساز youwave را به ویژوال استودیو وصل کنیم. برای این منظور دستور زیر را در cmd اجرا کنید.

```
adb connect localhost:5558
```

بعد از آن اگر پروژه را اجرا کنید، فایل apk. پروژه بر روی شبیه ساز نصب شده و اجرا خواهد شد. با کلیک بر روی دکمه‌ی confirm تصویری به شکل زیر قابل مشاهده خواهد بود:



علاوه بر این ما در سند HTML خود در بالا، یک page و یک تگ a قرار داده ایم.

```
<a href="#page2" data-transition="pop" data-rel="dialog" data-role="button" data-inline="true" data-icon="back">page 2</a>
```

data-role: با مقدار button در نظر گرفته شده است؛ لذا به شکل 4 دکمه دیگر رندر خواهد شد.
data-transition: با مقدار pop در نظر گرفته شده است که مشخص کننده ی افکت ظاهر شدن صفحه ای است که قرار است بار گذاری شود.

data-rel: مشخص می کند که صفحه ی مورد نظر من به صورت دیالوگ باز شود.
data-icon: با استفاده از این ویژگی می توان icon مورد نظر خود را برای المنت در نظر گرفت.

data-inline: برای به خط کردن دکمه ها کنار هم استفاده می شود.

با لمس کردن این دکمه، نتیجه به شکل زیر خواهد بود:



در مقاله‌ی بعد، به مباحث Database در Cordova خواهیم پرداخت.

ادامه دارد...

نظرات خوانندگان

نویسنده: افشین عباسپور
تاریخ: ۱۱:۵۲ ۱۳۹۴/۰۱/۱۱

خیلی خوب ... یک سوال :
آیا همیشه برای تست و اجرای برنامه از اندروید نصب شده روی نرم افزار VirtualBox در ویژوال استودیو استفاده کرد ؟
شما از نحوه انجام و تنظیمات برای اتصال به VitualBox اطلاع دارید ؟

نویسنده: غلامرضا ربال
تاریخ: ۱۴:۳۲ ۱۳۹۴/۰۱/۱۱

بنده این روش را تست نکردم . مرجعی هم در این باره نتواستم پیدا کنم . ولی حتما تست خواهم کرد .

نویسنده: محمد رضا صفری
تاریخ: ۲۲:۲۰ ۱۳۹۴/۰۱/۱۷

شماره 6 کی منتشر میشه ./ ؟

نویسنده: غلامرضا ربال
تاریخ: ۲۲:۳۷ ۱۳۹۴/۰۱/۱۷

ایشالا همین هفته وقت کنم حتما منتشر خواهم کرد.