فهرست مطالب

[مقدمه 6](#_Toc441913863)

[فصل اول :آشنایی با Apache Cordova و نصب و راه اندازی آن 8](#_Toc441913864)

[فرآیند بسته بندی 9](#_Toc441913865)

[درباره Web Views 9](#_Toc441913866)

[توضیح مختصری در مورد  Adobe Phonegap 12](#_Toc441913867)

[مزایای استفاده از Cordova: 12](#_Toc441913868)

[معایب  استفاده از Cordova : 13](#_Toc441913869)

[نصب اتوماتیک وابستگی ها 13](#_Toc441913870)

[نصب دستی وابستگی‌ها 15](#_Toc441913871)

[نکته تکمیلی 18](#_Toc441913872)

[رویداد‌های بومی 26](#_Toc441913873)

[روال کار jQuery Mobile 34](#_Toc441913874)

[افزونه های Cordova 35](#_Toc441913875)

[فصل دوم : آشنایی با فریم ورک Ionic 45](#_Toc441913876)

[معماری یک برنامه ترکیبی موبایل 45](#_Toc441913877)

[Apache Cordova چیست؟ 46](#_Toc441913878)

[Ionic چیست؟ 48](#_Toc441913879)

[نصب نرم‌افزارها 48](#_Toc441913880)

[نصب Node.js 48](#_Toc441913881)

[نصب Git 49](#_Toc441913882)

[نصب bower 50](#_Toc441913883)

[نصب Gulp 51](#_Toc441913884)

[نصب کُردوا و CLI آیونیک 52](#_Toc441913885)

[اولین برنامه با آیونیک 53](#_Toc441913886)

[فصل سوم : فصل سوم : اجزای CSS آیونیک و راهبری 59](#_Toc441913887)

[فصل 4: بکار گیری SCSS در پروژه آیونیک 64](#_Toc441913888)

[Sass چیست؟ 64](#_Toc441913889)

[راه‏اندازی scss در پروژه‏های آیونیک 66](#_Toc441913890)

[نصب دستی 67](#_Toc441913891)

[کار با CLI آیونیک 68](#_Toc441913892)

[کار با آیونیک scss 68](#_Toc441913893)

[تم پایه 68](#_Toc441913894)

[درک راه‏اندازی آیونیک scss 70](#_Toc441913895)

[خلاصه 75](#_Toc441913896)

[منابع 76](#_Toc441913897)

تقدیم به

همه دوستداران دنیای کد و توسعه

# مقدمه

در این کتاب یاد‌ ‌می‌گیریم که چگونه برنامه های موبایل را برای تمامی سکو های مختلف از جمله اندورید[[1]](#footnote-1)˛آی اُس[[2]](#footnote-2) ˛ ویندوز فون[[3]](#footnote-3) و بلک‌بری[[4]](#footnote-4) و سایر سکو های رایج با فناوری های سمت کاربر وب یعنی HTML,CSS,JavaScript توسعه دهیم. آنچه را که در این کتاب بررسی خواهیم کرد عبارت است از:

* فصل اول: در این فصل با نصب و راه‌اندازی Apache Cordova در ویژوال استودیو و نوشتن یک برنامه با نصب محیط شبیه ساز و افزونه های هشدار دهنده فصل را به پایان رسانده ایم.
* فصل دوم : با یکی از فریم‌ورک[[5]](#footnote-5) Ionic را به همراه وابستگی آن نصب خواهیم کرد و مروری بر مزیت های آن خواهیم داشت در نهایت یک برنامه را به کمک آن توسعه می دهیم.
* فصل سوم: به معرفی اجزای چارچوب آیونیک می پردازیم . مولفه‌هایی مثل سیستم گرید‌بندی‌˛دکمه ها‌˛منو ها و... را بررسی می کنیم و یاد‌خواهیم گرفت که چگونه رابط و تجربه کاربری بهتری را برای برنامه های موبایل ایجاد کنیم.
* فصل چهارم: استفاده از SASS شخصی سازی چارچوب آیونیک می پردازیم.

# فصل اول :آشنایی با Apache Cordova و نصب و راه اندازی آن

[**Apache Cordova**](http://cordova.apache.org/docs/en/4.0.0/) یک فریمورک سورس باز برای ساخت اپلیکیشن‌های **چند سکویی** موبایل (cross platform) با استفاده از Html5 می‌باشد.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:**

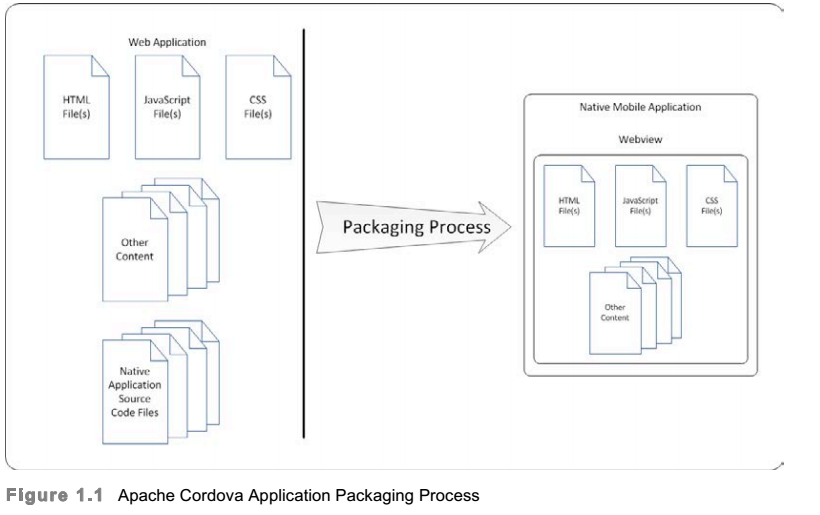
سورس کدی برای هر Container و برنامه محلی برای هر یک از سکوهای موبایل که پشتیبانی می‌شوند. container، کد‌های Html5 را بر روی دستگاه (Device) رندر می‌کند. (در مطالب بعدی در مورد این مطلب توضیح خواهم داد)

مجموعه‌ای از رابط‌های برنامه‌ی 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

[](http://www.dotnettips.info/file/image?name=1-5f69d98f9dd8402bb2e7cc14fb6d05f3.png)

عکس 1-1 فرآیند بسته بندی برنامه آپاچی کُردوا

 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 andoid.webkit.WebView) , در iOS با UIWebView موجود در (Using System/Library/Framworks/UIKit.framewor)  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 معمولا به اپلیکیشن‌های موجود در دستگاه، سخت افزار و یا API‌های بومی دستگاه، دسترسی ندارد. برای مثال شاید بگویید که یک وب اپلیکیشن معمولا به لیست مخاطبین با دوربین دستگاه و ... دسترسی ندارد.

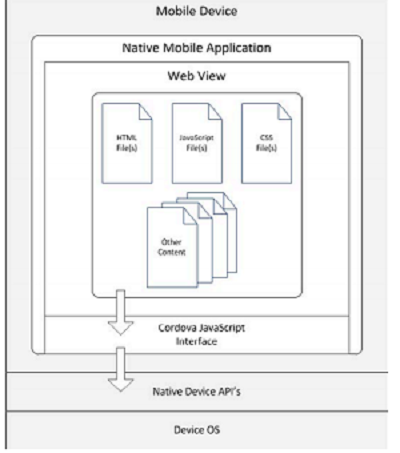
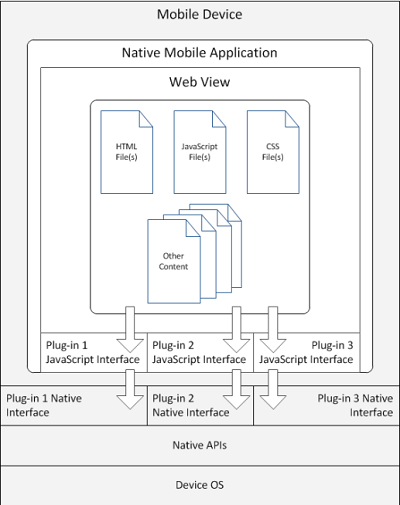
جواب : در واقع امکان دسترسی به این قابلیت‌ها توسط اپلیکیشن بومی (native mobile application) ایجاد می‌شود.

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 API‌ها در دو بخش پیاده سازی می‌شوند:**

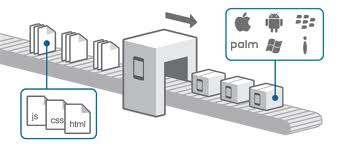
1-کتابخانه‌ی جاوااسکریپت که اجازه‌ی استفاده از قابلیت‌های بومی را به وب اپلیکیشن می‌دهد و کد بومی مشابه در Container اجرا می‌شود که مربوط است به بخش بومی این API ها. در اصل یک کتابخانه‌ی جاوا اسکریپت وجود دارد، اما بخش بومی API‌ها وابسته به سکوی (platform) انتخاب شده پیاده سازی می‌شود.

اگر شما از API‌های 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.

[](http://www.dotnettips.info/file/image?name=2-16d08383e46447cba570c37b2d740ee5.png)[](http://www.dotnettips.info/file/image?name=3-22b054ab66e74fcca659e78b33e20b55.png)

در قسمت قبلی ، با Apache Cordova  آشنا شدیم. در این قست قصد دارم در مورد Phonegap, معایب و مزایای Cordova و روش نصب و راه اندازی آن را بر روی Visual Studio، خدمت شما ارائه دهم.

عکس 1-2 پیاده سازی API های کردوا

[](http://www.dotnettips.info/file/image?name=phonegap-4-(1)-5a6d60fb327f4804beac754c151b8c51.jpg)

شکل 1-1 نمایش نحوه کارکرد آپاچی کُردوا

# توضیح مختصری در مورد  Adobe Phonegap

در حوالی سال 2009 ، phonegap به‌واسطه‌ی استارت آپی 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](https://en.wikipedia.org/wiki/Blink_(layout_engine)) و کروم است. در واقع phonegap ترکیبی‌است از Cordova و یک سری امکانات اضافه‌ی 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](http://ionicframework.com/blog/what-is-cordova-phonegap/) و [اینجا](http://www.dreamnotion.com/blog/development/mobile-app-development-phonegap-cordova-part1/)

## 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 HTML5 , CSS3
* برنامه نویسی آسان و سریع با javascript , Typescript
* قابلیت اجرا بر روی چندین پلتفرم مختلف(Android,iOS,Widnows Phone )
* قابلیت استفاده از فریم‌ورکهای تحت وب مانند Bootstrap , Angular JS, ...
* 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](http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/)

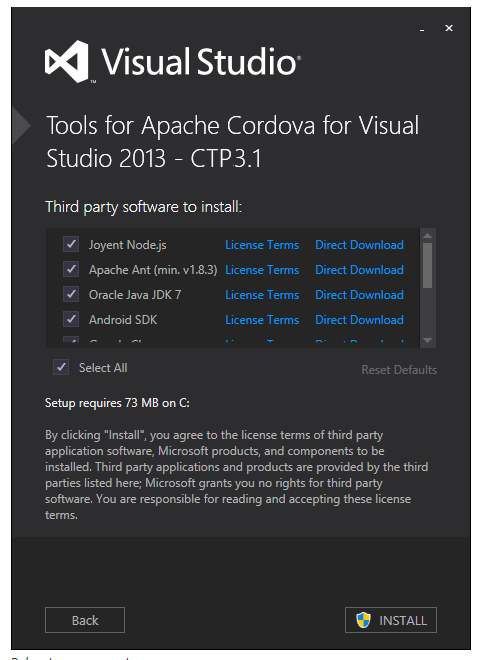
## معایب  استفاده از Cordova :

* نداشتن ابزار گزارش خطاهای مناسب؛ درنتیجه 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](http://www.dreamnotion.com/blog/development/mobile-app-development-phonegap-cordova-part1)

# نصب اتوماتیک وابستگی ها

ابزارهایی که ما نیاز داریم:

[](http://www.dotnettips.info/file/image?name=cord-1ca958390df74149afa2380a14aedf48.PNG)

1-2 باز نصب ویژوال استدویو برای نصب وابستگی ها کُردوا

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](http://www.microsoft.com/en-us/download/details.aspx?id=42675)

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 وابستگی ها:

* Node.js
* Git CLI
* Google Chrome
* Apache Ant
* Oracle Java JDK 7حتما نسخه x86 نصب شود.
* Android SDK
* SQLLite For Windows Runtime
* Apple iTunes

فایل نصاب، همه‌ی این وابستگی‌ها را به‌غیر از Android SDK، نصب می‌کند. [نسخه 32 بیتی](http://www.dotnettips.info/sdk%20x86%20http:/dl.ehsanavr.com/1393/06/adt-bundle-windows-x86-20140702.zip) و [نسخه 64 بیتی](http://www.dotnettips.info/post/2054/sdk%20x64%20http:/dl.ehsanavr.com/1393/06/jdk-7u67-windows-x64_EhsanAvr.Com.exe)

در آخر هم سیستم خود را راستارت کنید.

# نصب دستی وابستگی‌ها

اگر به هر دلیلی در نصب خودکار این وابستگی‌ها  توسط نصاب با مشکل بر خورد کردید، می‌توانید تک تک 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](http://go.microsoft.com/fwlink/?LinkID=396867)  (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](http://go.microsoft.com/fwlink/?LinkID=396869) 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 مشحص شده دانلود کنید. پکیچ‌های 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](http://websocket4net.codeplex.com/downloads/get/830268) ) 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

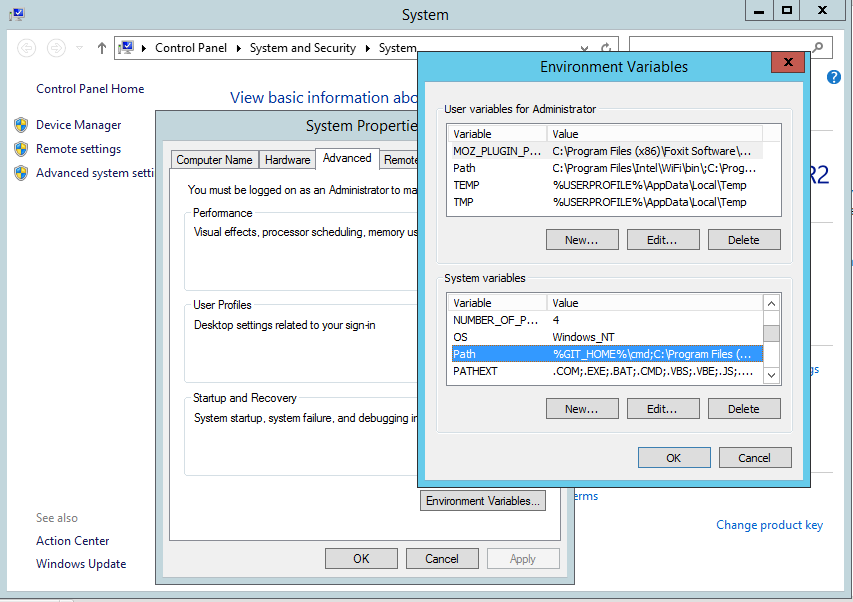
JAVA\_HOME: به مسیر نصب جاوا اشاره می‌کند

GIT\_HOME: به مسیر نصب GIT اشاره می‌کند.

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

برای تنظیم این متغیر‌ها، به مسیر Control Panel\System and Security\System وارد شده و گزینه‌ی Advanced System Setting را انتخاب کنید. سپس در پنجره‌ی باز شده گزینه‌ی Environment Variables را انتخاب کنید و در قست 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 متغییر 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 :

|  |  |
| --- | --- |
| 1  2 | %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; |

[](http://www.dotnettips.info/file/image?name=path.PNG) دقت کنید که مسیر‌های ذکر شده فقط یکبار در کل فایل وجود داشته باشند و سپس کل محتوای فایل را کپی

شکل 1-2 نحوه ثبت متغییر در ویندوز

کرده و در قسمت مربوط به path پیست کنید و با کلیک بر روی دکمه‌های OK کار را به اتمام رسانید.

# نکته تکمیلی

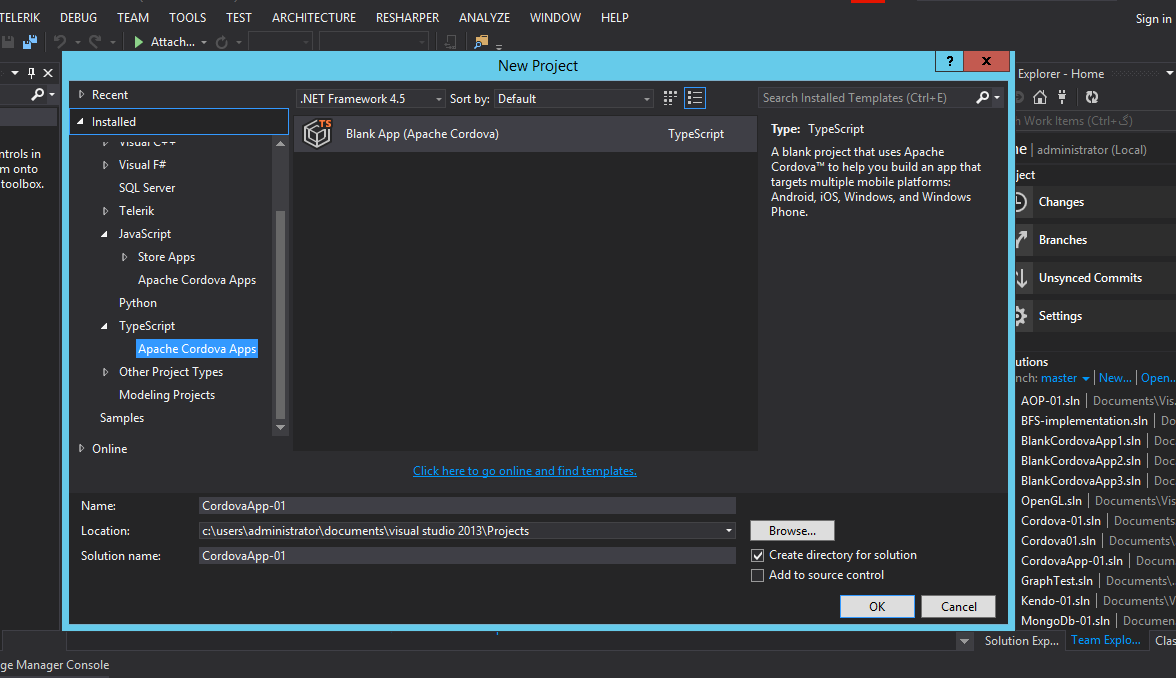
نیازمندی Apache Cordova CTP3.1   :

یکی از سیستم عامل‌های مقابل: Windows 7, Windows 8, Windows 8.1, or Windows Server 2012 R2.

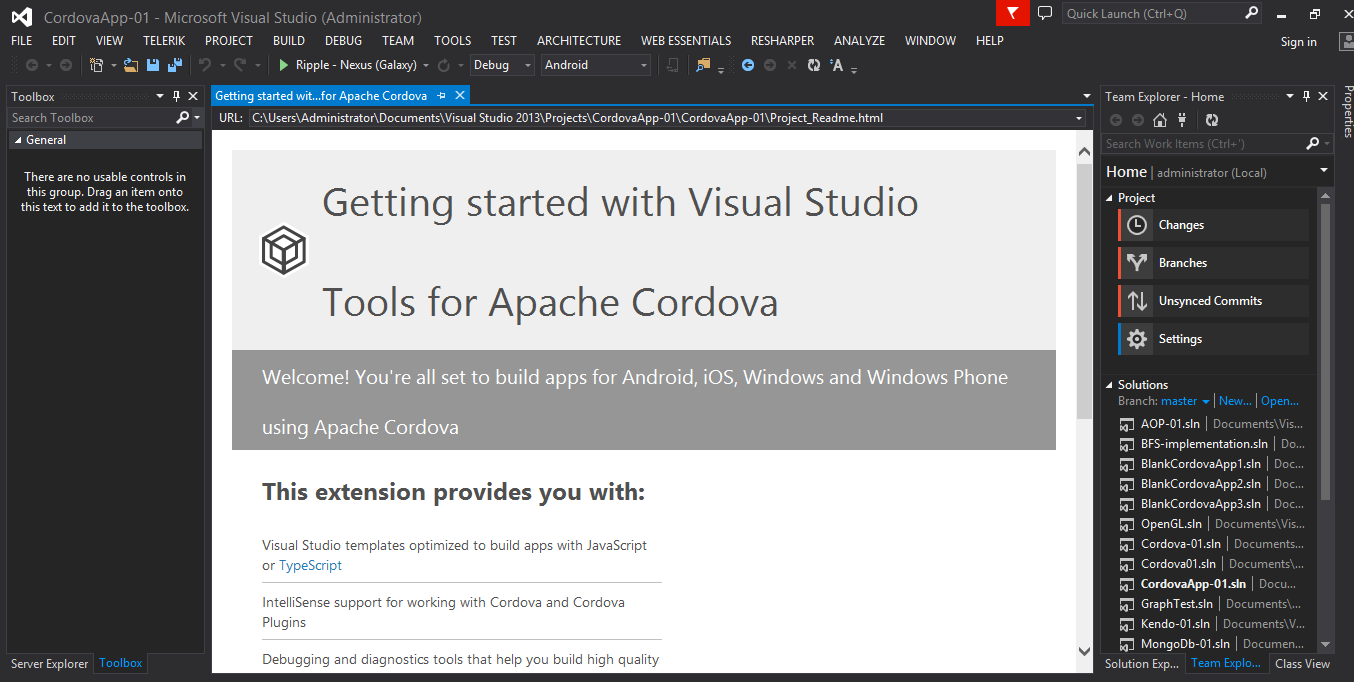
آپدیت 4 مربوط به ویژوال استدیو (دقت کنید قبل از نصب آپدیت 4 ویژوال استدیو باید نسخه قبلی  Cordova CTP  را حذف کنید(uninstall) )

امکان توسعه اپلیکیشن‌های windows phone , windows برای کاربران ویندوز 7 وجود ندارد .

[در قسمت قبل](http://www.dotnettips.info/post/2054/%d8%b4%d8%b1%d9%88%d8%b9-%da%a9%d8%a7%d8%b1-%d8%a8%d8%a7-apache-cordova-%d8%af%d8%b1-%d9%88%db%8c%da%98%d9%88%d8%a7%d9%84-%d8%a7%d8%b3%d8%aa%d9%88%d8%af%db%8c%d9%88-2) توانستیم ابزارهای لازم را برای Apache Cordova، نصب کنیم. در این قسمت یک پروژه‌ی ساده را ایجاد کرده و در مورد ساختار آن توضیح خواهم داد. در ادامه‌ی مقالات از  AngularJS ، Bootstrap ,Typescript و jQuery Mobile  هم در پروژه‌ها استفاده خوهیم کرد.

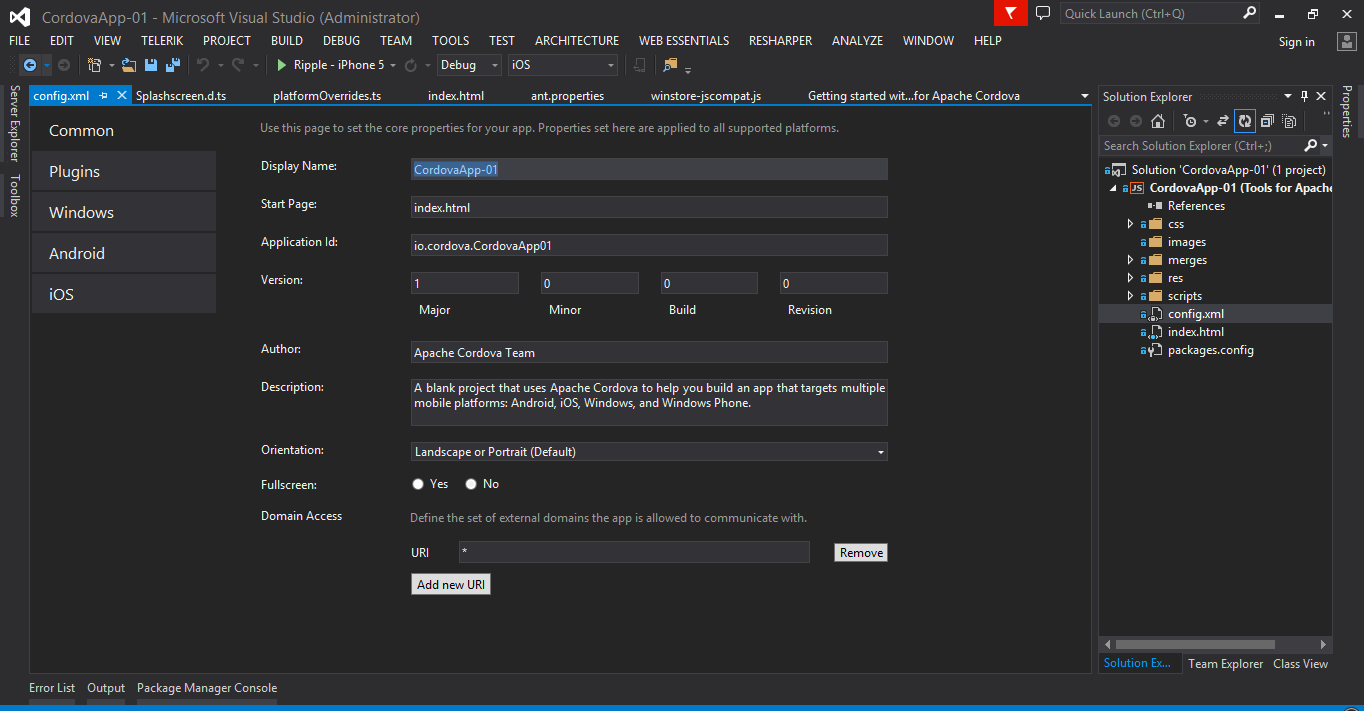
[](http://www.dotnettips.info/file/image?name=1-fd6c22808b3f4981824ab2a49b8f851a.png)برای شروع، از قسمت JavaScript یا Typescript، یک پروژه‌ی از نوع Blank App ایجاد کنید. به شکل زیر:

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](http://www.dotnettips.info/post/2054/%D8%B4%D8%B1%D9%88%D8%B9-%DA%A9%D8%A7%D8%B1-%D8%A8%D8%A7-apache-cordova-%D8%AF%D8%B1-%D9%88%DB%8C%DA%98%D9%88%D8%A7%D9%84-%D8%A7%D8%B3%D8%AA%D9%88%D8%AF%DB%8C%D9%88-2) به درستی صورت گرفته است.

[](http://www.dotnettips.info/file/image?name=2-3dd96da2744842c2a65d4a4fc7d53b0b.png)

1-3 باز کردن یک پروژه جدید کُردوا در ویژوال استودیو

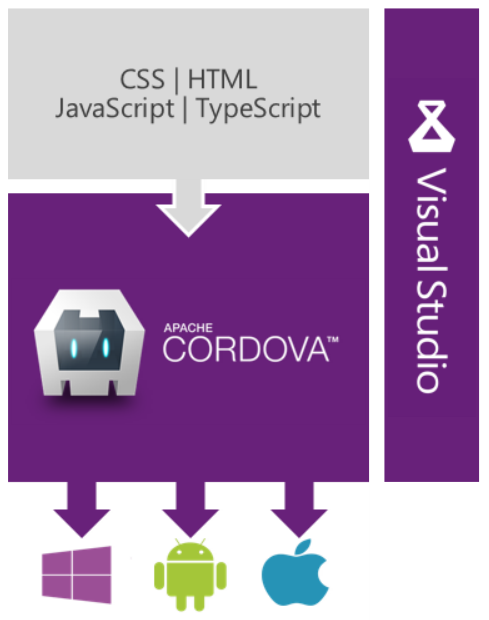
اگر به قسمت solution explorer دقت کنید، فایلی به نام config.xml را مشاهده خواهید کرد. با کلیک بر روی این فایل، یک صفحه‌ی گرافیکی باز خواهد شد که این امکان را به شما می‌دهد که پلاگین‌های مورد نیاز خود، تنظیمات مربوط به نرم افزار تولیدی (مانند تنظیم ورژن ویندوزی که می‌خواهید app شما بر روی آن اجرا شود)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

[](http://www.dotnettips.info/file/image?name=3-f9479ea4623443dcb3d008956086e40c.png)

یک فایل index.html هم در قالب پیش‌فرض قرار داده شده که بعدا 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 CSS ، HTML و JavaScript  نداریم و در ادامه‌ی مقالات با مثال‌های متعددی از آن‌ها استفاده خواهیم کرد.

در فولدر scripts-->typeings-->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.

[](http://www.dotnettips.info/file/image?name=cordova-architecture.png)

عکس 1-4 نحوه کار آپاچی کُردوا

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 API‌های 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:

|  |  |
| --- | --- |
| 1 | <script src="cordova.js"></script> |

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

|  |  |
| --- | --- |
| 1  2 | document.addEventListener("deviceready", onDeviceReady, false);  function onDeviceReady() { /\* INIT \*/ } |

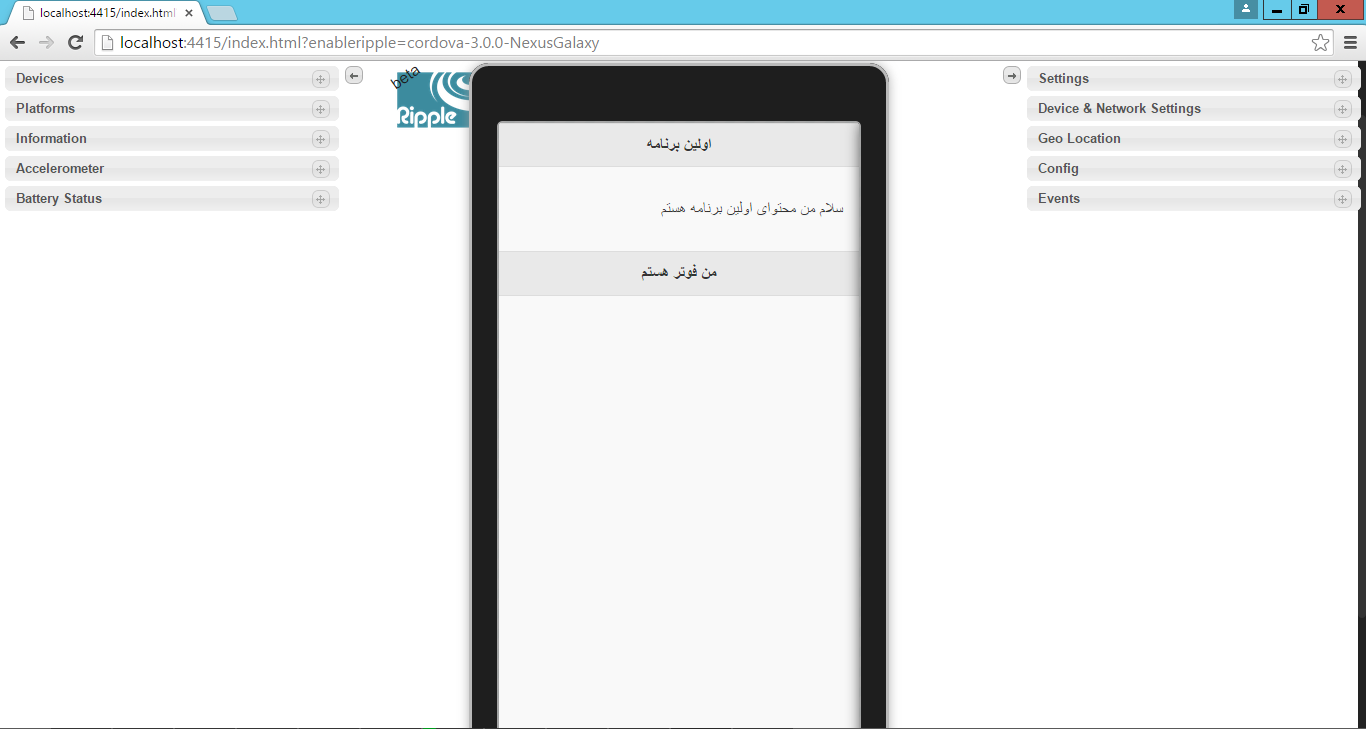
برای مدیریت رخدادهای مربوط به pause و resume هم که نشان دهنده‌ی ادامه برنامه (خارج شدن از حالت pause) و حالت تعلیق هستند، می‌توان به شکل زیر عمل کرد:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | 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 اجر ا 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](http://p30download.com/fa/entry/35086) 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

[](http://www.dotnettips.info/file/image?name=4-e978194cf84441d08bf7e7dbcb15657f.PNG)

**مطالعه بیشتر**

<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 ببنید 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 خواهم گرفت.

**توضیح تکمیلی در مورد ساختار فایل‌های پروژه**

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

|  |  |
| --- | --- |
| 1 | <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 همنام آن پلتفرم قرار دارد. برای مثال برای android، یک چنین دایرکتوری merges/android/scripts وجود دارد که درون آن فایلی به‌نام  platformOverrides.js دیده می‌شود و اگر دقت کنید، همنام فایل 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 "merges/"platformname ، در فولدر‌های هم نامی در شاخه‌ی ریشه‌ی پروژه کپی شده و جایگزین فایل‌های قبلی خواهند شد.

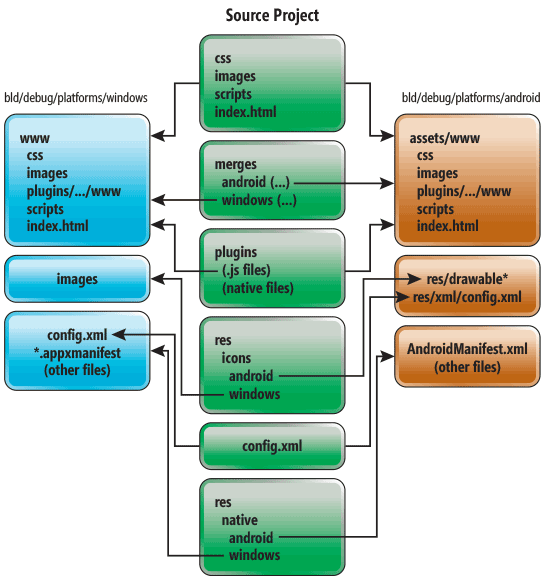
 **مثال برای اندروید**

در زمان ساخت (build) فایل scripts/platformOverrides.js با فایل merges/windows/scripts/platformoverrides.js جایگزین خواهد شد. این امکان برای فلدر‌های 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**

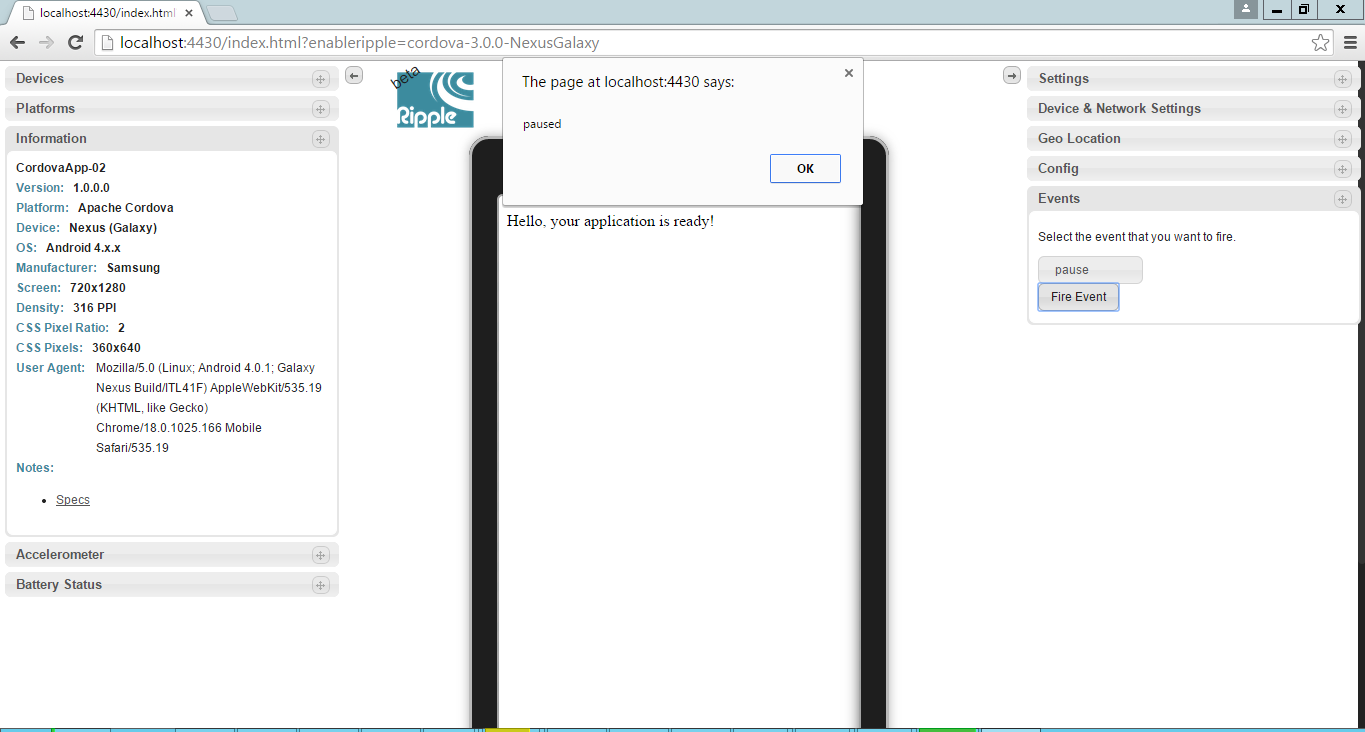
برای محتوای موجود در فولدر res، قضیه فرق می‌کند. زیرا محتوای این resource‌ها برای اپلیکیشن پکیچ ضروریست؛ پیش از آن که کد‌های ما درون WebView یا host رندر شوند. باید توجه کرد که این فولدر به جهت اینکه منابع اصلی را (با توجه به پلتفرم باید از فایل‌های مشخص آن برای تشخیص ساختار فولدر‌های اپلیکیشن پکیچ استفاده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

[](http://www.dotnettips.info/file/image?name=11-c57a347a5ce441e6afa97bdfb2c1b7fd.png)

# 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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | (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");      };      } )(); |

[](http://www.dotnettips.info/file/image?name=22-66baac5f1bd84021b76f50a8739174c8.PNG)

.در مقالات آینده از افزونه‌های موجود، برای مدیریت رخداد‌های باتری سیستم استفاده خواهیم کرد

**jQuery Mobile**

[جی کوئری موبایل](http://api.jquerymobile.com/)، 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 دهند گان 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 فایل های rtl.jquery.mobile-1.4.0.css و rtl.jquery.mobile-1.4.0.js موجود در زیر شاخه‌های فلدر مذکور را به head و آخر body فایل index.html اضافه کنید.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <!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 شده‌است.

نتیجه‌ی نهایی به شکل زیر خواهد بود:

[](http://www.dotnettips.info/file/image?name=Capture-e7b85ac1c18946a9bf06764f0e78a23a.PNG)

در مقاله‌ی بعد به استفاده از plugin‌ها خواهیم پرداخت.

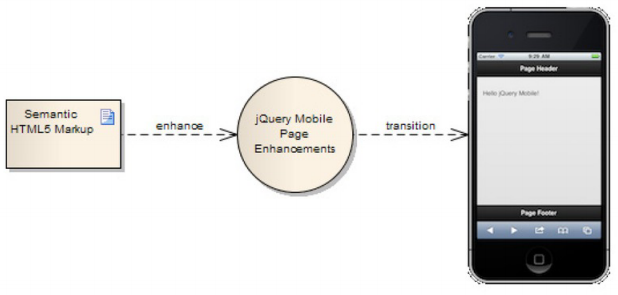
تگ متای زیر برای تنظیمات مربوط به viewport است و برای jQuery Mobile توصیه می‌شود.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!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 ویژگیdata-role علامت گذاری شده‌اند، CSS3 بهینه شده برای موبایل را اعمال می‌کند.

[](http://www.dotnettips.info/file/image?name=1-7a49d556b6574eb38ad28162f758cec0.png)

شکل 1-9 نحوه کار جی‌کوری موبایل

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 ویجتی خاص، با آن آشنا خواهیم شد.

[لیست کامل اتریبیوت‌های -data به همراه مقادیری که می‌پذیرند](http://api.jquerymobile.com/data-attribute/)

[دموی مربوط به ویجت‌ها](http://demos.jquerymobile.com/1.4.5/)

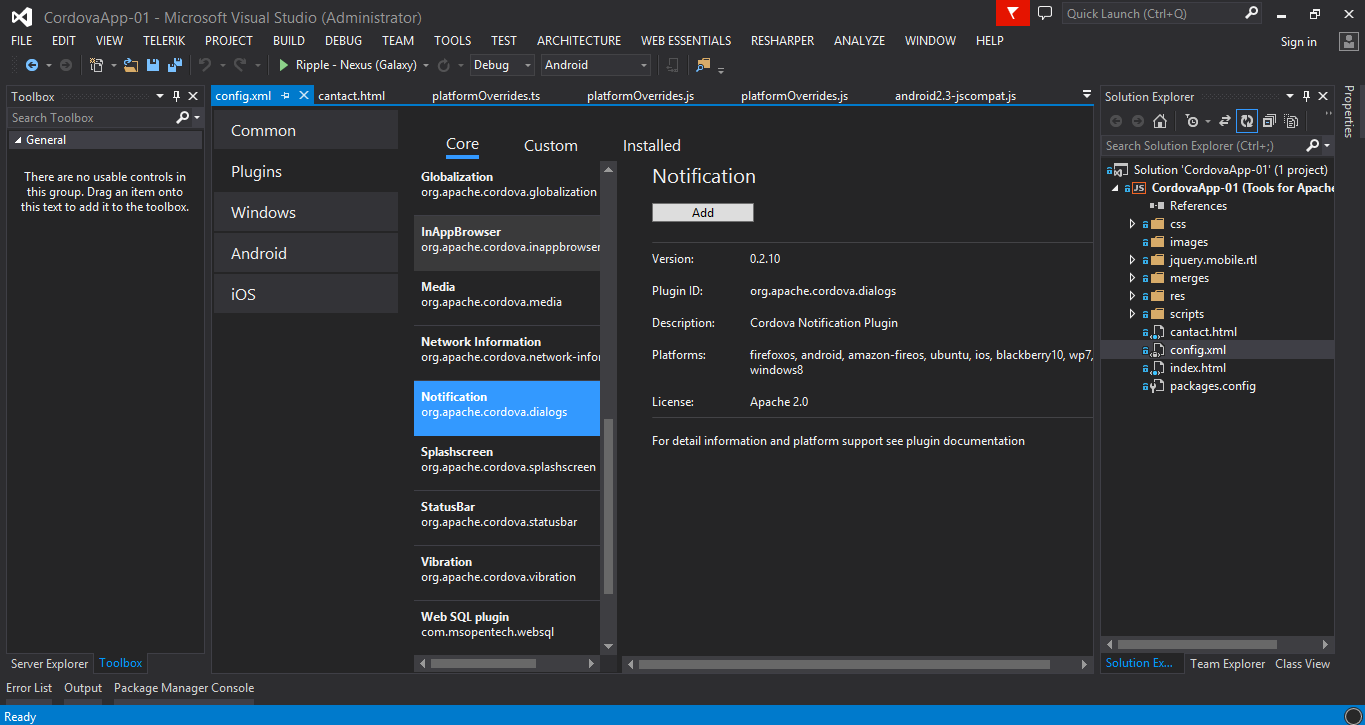
[لیست تمام رخدادها](http://api.jquerymobile.com/category/events/)

شما می‌توانید از امکانات [Theme Roller](http://themeroller.jquerymobile.com/)برای شخصی سازی تم‌های مورد نیاز استفاده کنید.

[لیست کامل کلاس‌های CSS](http://api.jquerymobile.com/classes/)

# افزونه های Cordova

از این قسمت <http://plugins.cordova.io/#/viewAll> و این قسمت [http://plugreg.com/plugins](http://plugins.cordova.io/#/viewAll) 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.

[](http://www.dotnettips.info/file/image?name=plugin.PNG)

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 , custome.js را به شکل زیر در نظر می‌گیرم:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | $(function() {      $("#alert").on('tap', function(event) {          navigator.notification.alert("اطلاعات ذخیره شد",null, "alert", "تایید");      });        $("#prompt").on('tap', function(event) {          navigator.notification.prompt("برای تائید نام خود را وارد کنید", onPrompt, "prompt", "تایید", "لغو"],"نام خود"]);      });        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", "نمیدانم"]);      });        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](http://plugins.cordova.io/#/package/org.apache.cordova.dialogs) آن را مطالعه کنید.

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <!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>              تست پلاگین Notification          </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.

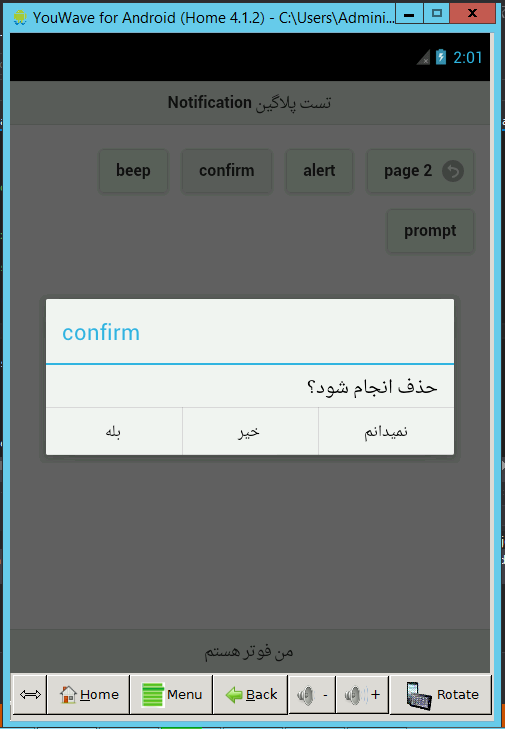
برای اینکار شبیه ساز [YouWave](http://p30download.com/fa/entry/35086/)را دانلود کرده و نصب کنید. سپس در قسمت toolbar ویژوال، گزینه‌ی Device را به جای شبیه ساز Ripple انتخاب کنید. نرم افزار youwave را اجرا کنید حال اگر برنامه را اجرا کنید با خطای زیر مواجه خواهید شد:

|  |  |
| --- | --- |
| 1  2  3  4 | Error447C:\Users\Administrator\Documents\Visual Studio 2013\Projects\CordovaApp-01\CordovaApp-01\bld\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 |

مشخصا 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.

|  |  |
| --- | --- |
| 1 | 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:

[](http://www.dotnettips.info/file/image?name=1-d2bb155c9a3046a09fac953b71c57e99.PNG)

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

|  |  |
| --- | --- |
| 1 | <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 کننده‌ی افکت ظاهر شدن صفحه‌ای است که قرار است بار گذاری شود.

data-rel: مشخص می‌کند که صفحه‌ی مورد نظر من به صورت دیالوگ باز شود.

data-icon: با استفاده از این ویژگی می‌توان icon مورد نظر خود را برای المنت در نظر گرفت.

data-inline: برای به خط کردن دکمه‌ها کنار هم استفاده می‌شود.

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

[](http://www.dotnettips.info/file/image?name=2-a20ba78ae74d4ca386c1de74345fae6c.PNG)

شکل 1-10 نمایش خروجی در محیط شبیه ساز

# فصل دوم : آشنایی با فریم ورک Ionic

در این فصل می‌خواهیم به یک دید کلی از برنامه های ترکیبی[[6]](#footnote-6) برسیم ˛ سپس نرم افزار های مورد نیاز برای توسعه برنامه های Ionic[[7]](#footnote-7) را نصب کنیم و در نهایت با کمک مولد کد[[8]](#footnote-8) چند برنامه را بسازیم.

موضوعاتی که آنها را در ادامه این فصل پوشش داده ایم:

* معماری برنامه های ترکیبی موبایل
* Apache Cordova چیست؟
* Ionic چیست؟
* نصب ابزارهای لازم برای توسعه و اجرای برنامه های Ionic
* کارکردن با قالب های Ionic
* استفاده از Yeoman برای ساختن خودکار برنامه های Ionic

# معماری یک برنامه ترکیبی موبایل

قبل از شروع کار با Ionic لازم است یک دید کلی از ساختار سکوی برنامه های ترکیبی موبایل داشته باشیم. مفهومی ساده و جالب دارد. تقریبا تمام سیستم عامل های موبایل (که هنگامی که با Cordova کار می‌کنیم سکو نامیده می شود) یک API[[9]](#footnote-9) برای توسعه برنامه‌هایش دارد. این API ها شامل مؤلفه‌ای است به نام **Web View**. این web view معمولاً یک مرورگر است که درون برنامه‌ی موبایل اجرا می‌‌شود. این مرورگر ها کدهای HTML,CSS را اجرا می کنند بدان معنی که شما صفحه را با استفاده از فناوری های وب توسعه و سپس آن را دورن برنامه خود اجرا می‌کنید.

شما با همان دانسته‌های توسعه وب می توانید برنامه های بومی-‌ترکیبی (در اینجا منظور از بومی فایل های قابل نصب مخصوص هر سکو روی دستگاه می‌باشد است) موبایل بسازید.

* اندروید از Android Application Package (**.apk**) استفاده می کند.
* iOS از iPhone Application Archive (**.ipa**) استفاده می‌کند.
* ویندوز فون از Application Package (**.xap**) استفاده می‌کند.

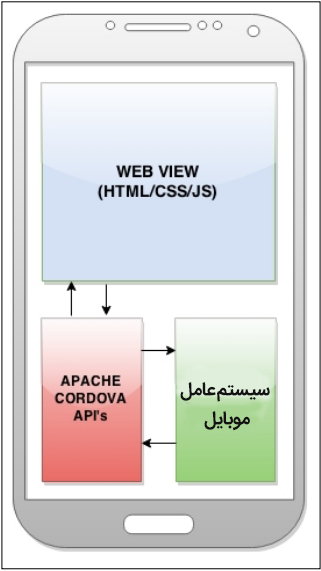
فایل بسته بندی شده یا نصب شونده شامل تکه کد‌هایی است که صفحه وب را راه انداری می کند و شاخه‌هایی از پوشه برنامه برای نشان دادن محتوای برنامه مورد نیاز است. به این نوع از راه اندازی صفحه وب درون فضای برنامه موبایل شامل منطق کاری برنامه شما می شود **برنامه های ‌ترکیبی** می گوییم.

# Apache Cordova چیست؟

به زبانی ساده Cordova یک نرم افزاری است که برنامه های کاربردی وب را با برنامه های کاربردی بومی موبایل پیوند می‌دهد. در [سایت](http://cordova.apache.org/) ‌‌‌Apache Cordova‌‌ آمده است که

« **آپاچی** کُردوا یک سکو برای ساخت برنامه های بومی موبایل با استفاده از HTML,CSS,JavaScript می‌باشد. »

آپاچی کُردوا فقط یک برنامه وب را به برنامه بومی پیوند نمی‌دهد ˛ بلکه یک سری API را که با جاوااسکریپت نوشته شده را برای کار کردن با ویژگی‌های بومی دستگاه را فراهم می‌کند. ‌شما می‌توانید با استفاده از جاوااسکریپت به دوربین دسترسی داشته باشید ˛ عکس بگیرید و به یک ایمیل ارسال کنید. هیجان انگیز به نظر می‌رسد نه؟

برای درک بهتر آنچه را که گفتیم اجازه دهید با هم به تصویر مقابل نگاهی بیاندازیم:

همان طور که می‌بینید ˛ ما یک web view داریم که‌ کد HTML,CSS,JavaScript در آن اجرا می شود. این کد می تواند صرفا قسمتی از رابط‌کاربری باشد. یا شما یک درخواست به صورت Ajax برای گرفتن داده ها به سمت یک سرور می‌فرستید ˛ یا کد می‌تواند بیشتر باشد مثلاً Bluetooth دستگاه را برای یافتن لیستی از دستگاه های مجاور فراخوانی کند.

عکس 2-1 نحوه ارتباط Apache Cordova با سیستم عامل موبایل

به عبارتی دیگر ˛ کوردوا یک سری API دارد که با web view توسط جاوااسکریپت ارتباط برقرار می‌کند. و با دستگاه با زبان بومی خودش صحبت می کند.(برای مثال Java برای اندورید) که کوردوا پلی بین شان فراهم می‌آورند. برای مثال اگر شما با جاوااسکریپت درباره برنامه در حال اجرای دستگاه اطلاعات بیشتری بخواهید ˛ فقط کافی است کد زیر را داخل فایل جاوااسکریپت بنویسید:

var platform = device.platform;

بعد از نصب افزونه‌ی دستگاه ˛ شما همچنین می توانید به UUID ˛ مدل ˛ نسخه سیستم عامل و ورژن کُردوای دستگاه از طریق web view با جاوااسکریپت به صورت زیر دسترسی داشته باشید:

var uuid = device.uuid;

var model = device.model;

var version = device.version;

var Cordova = device.Cordova;

در فصل های آینده در مورد افزونه‌های کُردوا بیشتر سروکار خواهیم داشت.

در توضیحات قبلی در مورد برنامه‌های ترکیبی موبایل و ساختار آنها حرف زدیم و هچنین به این اشاره کردیم که شما چگونه می توانید از ویژگی‌های دستگاه از راه web view با جاوااسکریپت استفاده کنید.

**نکته**: کوردوا کدهای HTML,CSS,JavaScript را به کد‌های مخصوص هر سیستم عامل تبدیل نمی‌کند. فقط کدهای HTML,CSS,JavaScript را بسته بندی می‌کند و آن را درون web view اجرا می‌کند.

**بنابراین ˛ شما باید حدس زده باشید که** Ionic یک چارچوب[[10]](#footnote-10) است که با آن ما کدهای HTML,CSS,JavaScript را می‌سازیم که بتواند در web view اجرا شود و با کردوا حرف بزند تا به API های مخصوص دستگاه دسترسی پیدا کند.

# Ionic چیست؟

**آیونیک** یک SDK[[11]](#footnote-11) نمای جلویی[[12]](#footnote-12) زیبا ˛ متن باز برای توسعه برنامه های ترکیبی با HTML5 می‌باشد.آیونیک اجزاء HTML,CSS ˛ انمیشن ها و ابزا‌های بهینه شده برای موبایل را برای ساختن برنامه های بسیار تعاملی فراهم می آورد.

آیونیک دارای کارا‌‌‌ئی بالایی با کمترین تغییر بر روی DOM و با کمترین انتقال سخت افزاری در مقایسه با سایر چارچوب های این زمینه را دارا می‌باشد. آیونیک از **AngularJS** به عنوان چارچوب جاوا اسکریپتی خود استفاده می کند.

با قدرت AngularJS درون یک چارچوبی مثل آیونیک تونایی ها نامحدود می‌شوند(شما می‌توانید از هر مؤلفه آنگولارجی‌اس درون آیونیک استفاده کنید در عین حال احساس یک برنامه موبایل به شما دست دهد). آیونیک با API کُردوای دستگاه به‌خوبی یکپارچه سازی شده است. این به این معنی است که شما می‌توانید با API های دستگاه با استفاده از کتابخانه‌ای مثل **ngCordova** دسترسی داشته باشید و آن را با مؤلفه های زیبای رابط‌کاربری آیونیک یکپارچه کند.

آیونیک **رابط خط فرمان**[[13]](#footnote-13) (CLI) خود را برای ساختن ˛ توسعه دادن و استقرار برنامه‌های آیونیک دارد. قبل از شروع کار با CLI آیونیک ما نیاز به نصب چندین نرم‌افزار داریم.

# نصب نرم‌افزارها

حالا ما می خواهیم قدم به قدم به نصب نرم‌افزار های لازم برای توسعه و اجرای برنامه های آیونیک بپردازیم.

## نصب Node.js

از آنجایی که آیونیک از Node.js برای CLI و انجام کار های خود بهره می‌برد ما آن را به صورت زیر نصب می کنیم:

1. به آدرس [https://nodejs.org](https://nodejs.org/)بروید.
2. بر روی دکمه **Install** در صفحه اول کلیک کنید فایل نصاب مربوط به سیستم عامل ‌تان بارگیری می‌شود.
3. Node.js را به وسیله فایل نصابی که بارگیری کرده اید نصب کنید.

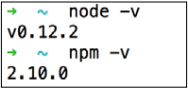
برای اینکه متوجه شوید که Node.js با موفقیت نصب شده است. دستورات زیر را در پایانه [[14]](#footnote-14)یا Prompt ویندوز بزنید.

*node -v*

که نسخه Node.js نصب شده را برمی‌گرداند و دستور زیر نیز نسخه npm که همراه با Node.js نصب می‌شود را بر‌می‌گرداند.

*npm -v*

شما باید چیزی شبیه این ببینید:



عکس 2-2 نمایش نسخه های npm,Node.js در سیستم عامل مکینتاش

npm کوتاه شده‌ی **Node Package Manager** است که ما برای بارگیری وابستگی های مختلف پروژه آیونیک از آن استفاده می کنیم.

نکته: شما به Node.js فقط در زمان توسعه نیاز دارید. نسخه های مشخص شده در تصویر قبلی صرفا جهت تکمیل توضیحات آورده شده است. شما ممکن است که نسخه ای مشابه یا آخرین نسخه نرم افزار را داشته باشید. و این برای تمامی عکس هایی که در این فصل نسخه های نرم‌افزار را نشان می‌دهد صادق است.

## نصب Git

Git یک سیستم کنترل نسخه توزیع شده[[15]](#footnote-15) رایگان و متن باز است که برای مدیریت تمامی پروژه های کوچک و بزرگ ˛ با سرعت و کارایی بالا طراحی شده است. در اینجا ما می خواهیم از یک Package Manager به نام Bower استفاده کنیم ˛ که از گیت برای بارگیری کتابخانه‌های مورد نیازش استفاده می‌کند.همچنین CLI آیونیک با استفاده از گیت قالب های خود را بارگیری می‌کند.

برای نصب گیت به آدرس <http://git-scm.com/downloads> بروید و فایل نصاب مربوط به سکوی خودتان را بارگیری کنید.بعد از نصب موفقیت آمیز آن به پایانه یا Prompt خود بروید و دستورات زیرا اجرا کنید:

*git –version*

که خروجی مشابه زیر را باید ببینید:



عکس 2-3 نمایش نسخه گیت در خط فرمان ویندوز

## نصب bower

ما می‌خواهیم از Bower (<http://bower.io/>) برای مدیریت وابستگی‌های کتابخانه‌ای برنامه‌مان استفاده کنیم.Bower نیز Package manager شبیهnpm است اما یک نسخه خطی یا تخت از آن است. این نوع از مدیریت بسته ها برای بارگیری منابع[[16]](#footnote-16) مورد نیاز توسعه وب مناسب تر هستند.

‌Bower برروی Node.js ساخته شده است. برای نصب به صورت سراسری از پایانه یا Prompt دستور زیر را اجرا کنید:

*npm install bower -g*

ما یک بسته node را به صورت سراسری نصب کردیم. در حالی که ممکن است پرچم -g در سیستم های بر مبنا \*nix عبارت sudo را نیاز داشته باشد ]در ویندوز باید خط فرمان را با حالت Run As Administrator باز کنید[.

اگر Bower با موفقیت نصب شده باشد پس از اجرای فرمان زیر باید خروجی مشابه عکس بگیرید:

*bower -v*



عکس 4-3 نمایش نسخه Bower در خط فرمان ویندوز

## نصب Gulp

حال Gulp (<http://gulpjs.com/>) را با هم نصب می کنیم ˛ که یک سیستم ساخت است که برروی Node.js توسعه پیدا کرده است. خودکار سازی تمامی کار های خسته کننده و تکراری توسط Gulp انجام می‌شود.

برای مثال زمانی که پروژه شما در آماده استفاده استفاده است ˛ شما ممکن است بخواهید که فایل های CSS,JavaScript,HTML خود را فشرده کنید˛ یا عکس هایتان را برای وب بهینه سازی کنید و کدهایتان را به محیط تولید [[17]](#footnote-17)ببرید. در این موارد Gulp ابزاری است که به کمک‌تان می‌آید.

تعداد زیادی افزونه برای گالپ وجود دارد که می‌تواند بیشتر کارهای تکراری را خودکار سازی کند˛ که با تلاش جامعه متن باز ساخته شده است. در آیونیک می‌خواهیم از گالپ برای تبدیل کد های SCSS به CSS استفاده کنیم.ما از کد SCSS برای شخصی سازی عناصر بصری آیونیک استفاده می کنیم.

برای نصب گالپ به صورت سراسری با خط فرمان دستور زیر را اجرا کنید:

*npm install gulp -g*

یا در سیستم عامل های بر پایه یونیکس این را اجرا کنید:

*sudo npm install gulp -g*

در نهایت پس نصب موفقیت آمیز گالپ با دستور زیر آن را بررسی کنید:

*gulp -v*

**

عکس 2-5 نمایش نسخه گالپ در خط فرمان ویندوز

## نصب کُردوا و CLI آیونیک

در نهایت برای نصب آیونیک ما CLI آیونیک را نصب می کنیم.CLI آیونیک همان CLI کُردوا است با چندین ویژگی اضافی می‌باشد.

نکته: تمامی کد مثال‌های این کتاب با استفاده از کُردوا نسخه 5.0.0 ˛ CLI‌ آیونیک نسخه 1.5.0 و آیونیک نسخه 1.0.0 (uranium-unicorn) می‌باشد.

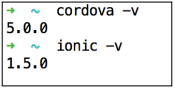
برای نصب CLI آیونیک دستور زیر را اجرا کنید.

*npm install cordova@5.0.0 ionic@1.5.0 -g*

برای اطمینان از نصب دستورات زیرا اجرا کنید:

*cordova –v*

*ionic –v*



عکس 2-6 نمایش نسخه کُردوا و آیونیک در ترمینال

برای اینکه داخل محتویات CLI آیونیک را ببینید˛ دستور زیرا را اجرا کنید:

*Ionic*

شما باید لیستی از وظایف را ببینید که در تصویر زیر آمده است:



عکس 2-7 نمایش لیست کارهای قابل انجام توسط آیونیک

**نکته**: تعداد بیشتری از وظایف وجود دارد که فقط تعدادی از آنها را در تصویر بالا نشان داده‌ایم.

می‌توانید توضیحات هر وظیفه را بخوانید برای اینکه ببینید چه کاری را انجام می‌دهد.و همچنین توجه کنید که یک سری از وظایف هنوز تا به امروز در حالت beta هستند.

با این کار ما نصب نرم‌افزار های مورد نیاز برای توسعه برنامه با آیونیک را کامل کردیم.

# اولین برنامه با آیونیک

هم‌اکنون ما نصب نرم‌افزار را انجام داده‌ایم˛ و می خواهیم چندین برنامه جدید را بسازیم.

آیونیک سه قالب آماده دارد که با استفاده از آن می توانیم توسعه برنامه های جدید را به‌‌‌سرعت شروع کنیم:

* **Blank**: این یک پروژه خالی آیونیک است با یک صفحه
* **Tabs**: یک برنامه ساده است که با استفاده از زبانه[[18]](#footnote-18)‌های آیونیک ساخته شده است
* **Side Menu**: این یک برنامه ساده است برای استفاده از منو کناری به منظور راهبری[[19]](#footnote-19) ساخته شده است.

برای درک پایه های سازنده‌˛ما از یک قالب خالی (‌Blank) شروع می کنیم.

برای اینکه فرایند یادگیری تمیز بماند˛ ما یک پوشه به نام ionicApps برای پروژه های آیونیک می سازیم و داخل آن یک پوشه‌ی دیگر برای فصل 2 با نام chapter2 ایجاد می نماییم.

سپس یک پایانه یا prompt جدید باز کنید و مسیر خود را به مسیر ionicApps و سپس به chapter2 تغییر دهید(cd[[20]](#footnote-20)).حال دستور زیر را اجرا کنید:

*ionic start -a "Example 1" -i app.example.one example1 blank*

در دستور قبلی :

* -a "Example 1" : نام قابل خواندن برنامه برای انسان است
* –i app.example.one : این شناسه برنامه یا معکوس نام دامنه است
* example1 : این نام پوشه است
* blank : این نام قالب است

CLI آیونیک وقتی که بخواهد کاری را انجام دهد مدت زمان نسبتا زیادی طول خواهد کشید. همان‌طور که در پایانه یا prompt می‌بینید˛ ‌در زمانی که پروژه ساخته می‌شود اطلاعات زیادی را چاپ می‌کند.

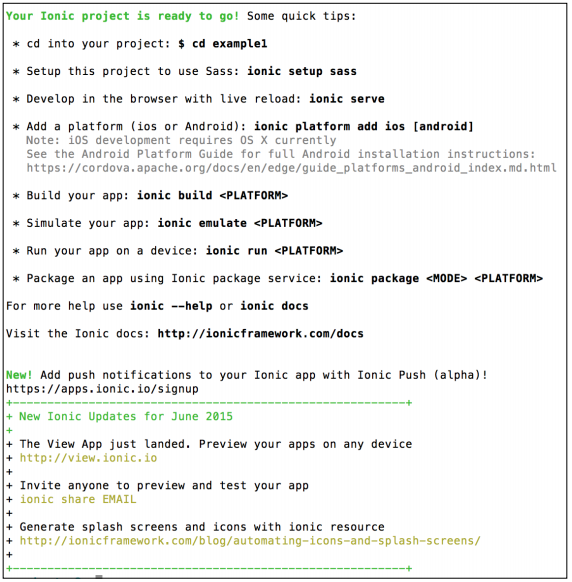
با شروع‌˛‌یک پروژه خالی بارگیری می‌شود و داخل پوشه example1 ذخیره می‌شود. سپس‌˛ ionic-app-base از مخزن https://github.com/driftyco/ionic-app-base ˛GitHub˛ ionic-app-base بارگیری می شود.در پایان ionic-startertemplate از مخزن GitHub با آدرس <https://github.com/driftyco/ionic-starter-blank> بارگیری می‌شود.

بعد از آن‌˛ فایل پیکربندی با شناسه برنامه به‌روز‌رسانی می‌شود. سپس یک اسکریپت اجرا می‌شود و پنج افزونه کُردوا بارگیری می‌شوند:

* org.apache.cordova.device (<https://gitHub.com/apache/cordovaplugin-device>) : این برای گرفتن اطلاعات دستگاه همان طور که قبلاْ در این فصل مشاهده کردید استفاده می‌شود.
* org.apache.cordova.console (<https://gitHub.com/apache/cordovaplugin-console>) : این افزونه برای استفاده از مزایای console.log() میانجی گری می کند.
* cordova-plugin-whitelist (<https://github.com/apache/cordovaplugin-whitelist>): این افزونه لیست سفیدی از سیاست های راهبری برنامه web view روی کُردوای نسخه 4.0 پیاده سازی می‌کند.
* cordova-plugin-splashscreen (<https://github.com/apache/cordovaplugin-splashscreen>) : این افزونه صفحه آغازین را در حین بالا آمدن برنامه نمایش می‌دهد و پنهان می‌کند.
* com.ionic.keyboard (<https://gitHub.com/driftyco/ionic-pluginskeyboard>) : این یک افزونه برای صفحه‌کلید است که کار کردن با آن را آسان‌تر می‌کند.و محرک رویداد هایی است که باعث نمایش یا پنهان شدن صفحه‌کلید می‌شود.

تمامی این اطلاعات نیز به فایل package.json اضافه می‌شود و یک فایل ionic.project جدید نیز ساخته می‌شود.

زمانی که پروژه با موفقیت ساخته شود‌˛تعدادی دستورالعمل اضافی را می‌بینید که چگونه‌ کار بیشتری را انجام دهید. شما باید خروجی مشابه عکس زیر را ببینید:



عکس 2-8 خروجی آیونیک‌˛ هنگام نصب موفقیت‌آمیز قالب پروژه

بعد از تغییر مسیر پروژه به پوشه example1‌˛ با دستور زیر پروژه خود را روی سرور اجرا کنید:

*ionic serve*

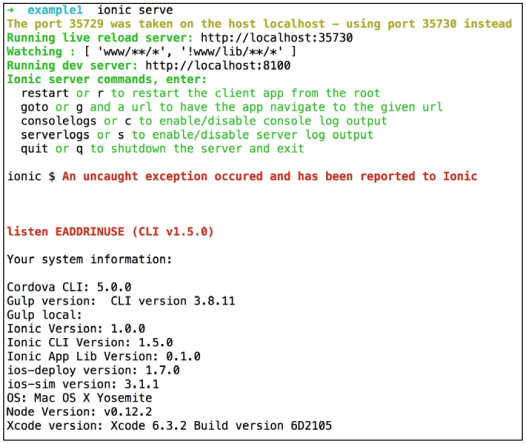
این سرور را روی پورت 8100 شروع بکار می‌کند و برنامه برروی مرورگر پیش‌فرضتان شروع به کار می کند.پیشنهاد می‌کنیم که Google Chrome یا Mozilla Firefox را عنوان مرورگر پیش‌فرضتان در زمان کار با آيونیک انتخاب کنید.

وقتی مرورگر بالا می‌آید˛ شما باید یک قالب خالی را مشاهده کنید.

اگر شما این را اجرا کنید :

*ionic serve*

پیام خطائی که در زیر می‌بینید مشاهده خواهید کرد:



عکس 2-9 خطای ناشی از اجرا شدن برنامه‌ای ‌دیگر روی پورت 8100

این به این معنی است که برنامه‌ای دیگر روی پورت 8100 در حال اجرا است. برای حل این مشکل شما می‌توانید از پورتی دیگر مانند 8200 برنامه خود را اجرا کنید:

*ionic serve –p 8200*

بعد از اجرای موفقیت آمیز برنامه و بالا آمدن ˛ خروجی آن را می‌توانید در مرورگر ببینید. به پایانه یا Prompt برمی‌گردیم.

# فصل سوم : فصل سوم : اجزای CSS آیونیک و راهبری

در قسمت های قبلی دیدیم که Ionic چیست و در کجای دنیای برنامه نویسی چندگانه موبایل قرار می گیرد.ما 2 راه برای چارچوب بندی اپ های ساخته با Ionic داریم. راه اول استفاده از Ionic CLIاست و روش دوم بهره گیری از استفاده از سازنده آیونیک(Ionic generator) است.در این فصل ما با مولفه های CSS که در Ionic موجود هستند ، سرو کار خواهیم داشت و با سیستم grid آیونیک و state router آشنا خواهیم شد .همچنین نگاهی خواهیم داشت بر مولفه ها و جزئیات مختلف Ionic به این منظور که بهره گیری از کدامیک از آن ها ما را به ساخت یک اپ با محیط و تجربه کاربری خوب خواهد رساند.

این فصل موارد زیر را پوشش خواهد داد:

* سیستم گرید Ionic
* جزئیات و مولفه های CSS
* ادغام و یکپارچه سازی ترکیبات CSS با AngularJS
* Ionic state router

مولفه های CSS آیونیک

در واقع آیونیک مجموعه ای از یک فریم ورک CSS قدرتمند به منظور برنامه نویسی موبایل و بسیاری از سرویس های AngularJS است.به وسیله این امکانات زمان لازم برای تبدیل یک ایده به یک اپلیکیشن در کمترین مقدار خود می رسد.

سیستم اندازه گیری(grid) آیونیک

برای داشتن کنترل کافی روی اجزای برنامه و نحوه چینش و قراگیری آن ها در کنار هم به یک سیستم قدرتمند grid نیاز است که Ionic راه حل خوبی است.

مزیتی که سیستم grid آیونیک دارد این است که مبتنی بر flexbox است.flexbox (CSS Flexible Box Layout Module ) در واقع برای بهبود و افزایش کارایی طراحی رابط کاربری یک box Module را در اختیار برنامه نویس قرار می دهد.

مزیتی که سیستم مبتی بر flex box دارد این است که دیگر شما نیازی به سیستم gridستونی ندارید. شما قادر هستید که هر چند تا ستونی را که میخواهید داخل سطر ایجاد کنید که در انتها همه آن ها به صورت خودکار در یک عرض قرار خواهند گرفت. بر خلاف سایر سیستم های grid مبتنی بر CSS دیگر لازم نیست نگران مجموع اسم کلاس هایی که در حال افزوده شدن هستند باشید.

برای اینکه به درک عمیق تر ی از سیستم grid دست پیدا کنید ، فایل index.htmlرا که در فولدر example5/www قرار دارد . در بین دایرکتیو ion-content کد زیر را اضافه کنید:

برای اینکه به صورت تصویری تفاوت را به عینه ببینید کد زیر را به تگ <head> اضافه کنید:

فایل index.html را ذخیره کنید. با استفاده دستور cd به فولدر example5 و مورد زیر را اجرا کنید.

*Ionic serve*

بعد مشاهده می کنید که:

برای اینکه چک کنیم که عرض به صورت خودکار کاهش می یابد ، تعداد divهای فرزند را به 3 کاهش می دهیم. همانطور که در قسمت پایین مشاهده می کنید:

و که در صورت اجرا به صورت زیر نمایش داده می شود.

بدون هیچ گونه زحمت و بدون نیاز به هرگونه شمارشی، تمام آنچه نیاز است شما انجام دهید این است که col(ستون)هایی را که نیاز دارید اضافه کنید. آن ها به صورت خودکار در یک عرض قرار خواهند گرفت.

قابلیت بالا به این معنی نیست که شما امکان درج مقدار عرض به صورت دستی و نابرابر با هم را ندارید.به کمک کلاس های موجود در Ionic این امکان برای شما فراهم است.

برای مثال تصور کنید که برای درصد دهی عرضهای ستون ،بخواهیم اینگونه عمل کنیم که ستون اول 50 درصد و دو ستون باقی مانده در 50درصد باقی مانده قرار بگیرند.برای این کار کافیست یک کلاس به نام col-50در ابتدای div به صورت زیر اضافه کنیم:

سپس باید نتیجه ای مشابه شکل زیر مشاهده کنیم:

همچنین میتوانید از کلاس هایی که از قبل تعریف شده اند استفاده کنید. در جدول زیر لیستی از این کلاس ها را شاهده می کنید.

بعد از colمیتوانید با افزودن هر یک از کلاس هایی که در جدول بالا به آن ها اشاره شد به مقدار مشخصی از عرض دسترسی پیدا کنید.

همچنین میتوان یک ستون با یک انحراف عرض مشخص را به برنامه افزود:

سپس با اجرای کد شکل زیررا مشاهده خواهید کرد:

Div اول دارای انحراف 33درصدی میباشد. 66درصد باقی مانده نیز بین 2 div تقسیم خواهند شد. در واقع کلاس های offset یک padding را د ر سمت چپ div ایجاد میکند.

در جدول زیر نیز تعدادی از کلاس های آماده offset نیز نشان داده شده است:

شما میتوانید ستون ها را به صورت عمودی نیز قرار دهید.این امکان مزیت دیگر flex box به شمار می رود.

کد زیر را نیز به موارد قبلی اضافه کنید:

نتیجه به صورت زیر نمایش داده می شود:

اگر قسمت سر یا بالایی یک از ستون ها از سایرین بلند تر باشد می توانید با استفاده از کلاس col-top میتوان محتوای ستون ها را در موقعیت دلخواه قرار داد. با افزودن کلاس col-center محتوای در قسمت وسط قرار می گیرد.همچنین کلاس col-bottom محتوای ستون را در قسمت پایینی سطر قرار می دهد.

ساختار صفحه

در ادامه مطالبی را در مورد ساختار صفحه یاد میگیریم که در ساخت اپ های تک صفحه ای با آیونیک بسیار کار آمد خواهد بود.

برای شروع ساخت یک اپ خالی مراحل زیر را دنبال کنید:

Ionic start –a “example 6” -1 app.six example6 blank

با استفاده از دستور cd به فولدر example6 مراجعه کنید و دستور زیر را اجرا کنید:

Ionic serve

با اجرا مورد بالا شما شاهد اجرای یک اپ خالی در مروگر پیش فرض خود خواهید بود.

فایل example6/www/index.html را در ویرایشگر مورد نظر خود اجرا کنید. در داخل تگ body باید دستوراتی شبیه به تکه کد زیر ببینید:

همه صفحه به صورت wrap در داخل یک دایرکتیو ion-pane قرار گرفته اند.

Ion-pane(<http://ionframewok.com/docs/api/derective/ion> pane/) یک container ساده است که محتوایات را در یک نما به صورت متناسب قرار داده است.

در ادامه می توانید دایرکتیو ion-header-bar([http://ionicframework.com/docs/api/directive /ionPane/](http://ionicframework.com/docs/api/directive%20/ionPane/)) ببینید. این دایرکتیو یک header ثابت را به صفحه اضافه می کند.توجه دشته باشید که attribute های کلاس به دایرکتیوion-header-bar افزوده می شود.

تصویر پایین نیز نشان دهنده ی رنگ های مشخصی است که مشخص کننده ی حالات مشخصی هستند:

کلاس bar-stable به دایرکتیو ion-header-bar افزوده شده است.با تغییر قسمت stable کلاس ، میتوان از سایر هدر های آماده استفاده کرد.

برای مثال به جای stable، assertive قرار دهیم، پس زمینه header به شکل زیر نمایش داده خواهد شد:

در قسمت های بعدی به مباحث دیگری در زمیه رنگ به کمک SCSS خوهیم پرداخت.

دایرکتیوی که بعد از header مورد بررسی قرار می گیرد. دایرکتیو ion-content است.ion-content این امکان را برای شما فراهم میکند که فضایی را برای قراردادن محتوا تعریف کنید که overflow نشود. این امکان در جاهایی که به اسکرول شدن محتوا نیاز است بسیار مفید و کاربردی است. البته در این مورد میتوان از $ionicScrollDelegate نیز استفاده کرد . که در فصل 5 که مربوط به مشتقات و سرویس های Ionic است بیشتر آشنا خواهیم شد.

برای اینکه ساختار کاملی برای یک صفحه داشته باشیم ، نیاز است که یک footer نیز برای آن تعریف کنیم . قبل از پایان دایرکتیو on-pane تکه کد زیر را اضافه کنید:

حال فایل را ذخیره کنید.آنچه در مرورگر خود مشاهده می کنید باید چیزی مثل تصویر زیر باشد:

بنابراین دیدیم که اگر بخواهیم یک اپ تک صفحه ای را با ionic بسازیم ، به ساختاری مشابه دستورات زیر نیاز مندیم:

اگر بخواهیم این دستورات را بدون استفاده از دایرکتیو های بنویسیم ، به شکل زیر عمل خواهیم کرد:

به علاوه شما میتوانید به HEADER وfooter خود ، دکمه هایی را هم اضافه کنید. دستورات زیر نحوه افزودن دکمه را نشان می دهد:

دکمه ای که قبل از تگ h1 در دارکتیو ion-header-barافزوده می شود در قسمت سمت چپ h1نمایش داده می شود.و به همین شکل دکمه ای که بعد از h1 افزوده میشود در سمت راست آن نمایش داده خواهد شد. شکل زیر گویای این توضیحات است:

توجه کنید که میتوان به روش مشابه دکمه هایی را در قمست footer نیز اضافه کرد.

دایرکتیوion-header-bar یکی از روش هایی که میتوان با آن header هایstatic ساخت.اما وقتی که صحبت از ساخت یک اپ چند صفحه ای در میان باشد ، باید چاری برای header ها کرد .با استفاده از دایرکتیو ion-nav-bar میتوان هدر هایی با کارآیی بیشتر ایجاد کرد. به این موضوع در بخش مربوط به state router ها خواهیم پرداخت.

# فصل 4: بکار گیری SCSS در پروژه آیونیک

در این فصل، نگاهی به استایل‏های ظاهری یک برنامه آیونیک می‏اندازیم. به طور پیش فرض دیدیم که آیونیک 7 سایه دارد، یا قالب با رنگ‏های از پیش تعریف شده‏اند. در این فصل ما به تغییر رنگ‏ها و تغییر اجزا آیونیک می‏پردازیم. این فصل به شما می‏گوید که در واقع چگونه آیونیک SCSS را به طور کلی درک کنیم به جای تمرکز بر اجزای خاص.

در این فصل، ما به ویرایش این رنگ‏ها و تغییر نگاه به اجزای آیونیک.

در این فصل به پوشش موضوعات زیر میپردازیم:

* Sass در مقابل scss
* راه‏اندازی scss
* کار با متغییر‏های scss
* کار با mixinهای scss
* قالب‏دهی side menu app

# Sass چیست؟

بر طبق مستندات sass در سایت http://sass-lang.com/documentation

" sass گسترش یافته css است که قدرت و ظرافت را به زبان پایه می‏افزاید. که به شما اجازه می‏دهد از متغییرها، قوانین تو در تو، mixinها، ایمپورت کردن و ... استفاده کنید. که همه اینها به طور کامل با قوانین css سازگار است. به کمک sass میتوان stylesheet های بزرگ را سازماندهی کرد و یک stylesheet کوچکتر با اجراع سریعتر را بدست آورد."

به عبارت ساده‏تر، sass css را قانونمند می‏کند. شما ممکن است تعجب کنید که چرا این فصل scss نامیده شده در حالی که ما درباره Saas صحبت می‏کنیم. Sass و Scss تقریبا یکسان و نوعی پیش پردازنده CSS هستند. با داشتن هریک راهی را برای نوشتن CSS پیش پردازش شده دارید.

Sass تکامل یافته یک پیش پردازنده دیگر به نام HTML است که به وسیله توسعه دهندگان روبی به وجود آمده است. بنابراین، بسیاری از خصلت‏های قوانین روبی را به ارث برده است. مانند تو رفتگی‏ها، نبود براکت، نبودن سمی کالن و ... .

به عنوان نمونه فایل Sass شبیه مثال زیر است:

// app.sass

brand-primary= blue

.container

color= !brand-primary

margin= 0px auto

padding= 20px

=border-radius(!radius)

-webkit-border-radius= !radius

-moz-border-radius= !radius

border-radius= !radius

\*

+border-radius(0px)

هنگامی که شما یک کد Sass رو با یک کامپایلر Sass اجرا می‏کنید. آن را یه حالت CSS قدیمی باز می‏گرداند. به مانند دستورات زیر:

.container {

color: blue;

margin: 0px auto;

padding: 20px;

}

\* {

-webkit-border-radius: 0px;

-moz-border-radius: 0px;

border-radius: 0px;

}

اما اگر شما به brand-primary در کدهای Sass توجه کرده باشید در عمل مانند یک متغییر عمل می‏کنند، جایگذاری آن مقدار در داخل کلاس container یا در border-radius به عنوان یک تابع عمل می‏کند(mixin نامیده می‏شود)، تولید قوانین CSS هنگامی که آرگومان صدا زده می‏شود این در CSS وجود نداشت.

کسانی که از کد نویسی زبان‏هایی که بر مبنایه براکت هستند استفاده می‏کنند این روش کد نویسی سخت است. بنابراین به SCSS وارد می‏شویم.

// app.scss

$brand-primary: blue;

.container{

color: !brand-primary;

margin: 0px auto;

padding: 20px;

}

@mixin border-radius($radius) {

-webkit-border-radius: $radius;

-moz-border-radius: $radius;

border-radius: $radius;

}

\* {

@include border-radius(5px);

}

به نظر می‏رسد بسیار شبیه CSSاست، درسته؟ و این بسیار واضح‏تر است. بهترین قسمت این جریان استفاده آیونیک از SCSS برای قالب دهی اجزاری آن است.

حالا ما درک پایه‏ای از اینکه SCSS و Sass چه چیزی هستند داریم، ما آن‏ها را در برنامه‏های آیونیک و قالب برای اجزا آن بررسی می‏کنیم.

# راه‏اندازی scss در پروژه‏های آیونیک

حالا ما طریقه نصب scss در پروژه‏های آیونیک آماده را می‏بینیم. یک فولدر با نام chapter4 ایجاد کرده و در داخل آن ترمینال را باز میکنیم و دستور زیر را اجرا میکنیم:

ionic start -a "Example 13" -i app.example.thirteen example13 tabs

به دو روش می‏توانیم scss را در پروژه‏ نصب کنیم:

* روش دستی
* کار یا CLI آیونیک

## نصب دستی

برای نصب scss به صورت دستی، گام‏های زیر را دنبال می‏کنیم:

1. از دستور cd، برای ورود به فولدر example13 استفاده می‏کنیم:  
   cd example13
2. پیشنیازهای لازم را نصب می‏کنیم. پروژه آیونیک که ما قالب‏ها می‏سازیم با یک فایل .json همراه است. این فایل پیشنیازای لازم برای نصب scss را به همراه دارد. بنابراین، ..
3. برای نصب پیشنیازها، دستور زیر را اجرا کنید:  
   npm install
4. همچنین، اگر Gulp هم نصب نیست، شما میتوانید با دسنور زیر نصبش کنید:  
   npm install gulp –global
5. حالا فایل www/index.html را باز کنید و خط توضیحی را در تگ هد بیابید که شبیه نوشته زیر باشد:  
   <!-- IF using Sass (run gulp sass first), then uncomment below and remove the CSS includes above <link href="css/ionic.app.css" rel="stylesheet"> -->  
   آن را از حالت توضیح خارج کنید. حالا ارجاع به فایل ionic.css را از بالای رفرنس‏ها پاک کنید. کار دیگه‏ای لازم نیست.
6. به ترمینال برگردید و دستور زیر را اجرا کنید:  
   gulp sass  
   و این دستور دو فایل ionic.app.css و ionic.app.min.css را در داخل پوشه www/css ایجاد می‏کند.

این تمام چیزی است که شما در نصب scss در پروژه آیونیک نیاز دارید. بعداً نگاهی هم به شخصی سازی scss خواهیم داشت.

## **کار با CLI آیونیک**

حالا نگاهی هم به نصب scss با استفاده از Ionic CLI's setup task داریم. قبلا ما scss را مثال 13 نصب کردیم. یه پروژه جدید با استفاده از CLI task می‏سازیم.

برای ساختن پروژه جدید، دتور زیر را اجرا می‏کنیم:

ionic start -a "Example 14" -i app.example.fourteen example14 tabs

بعد دایرکتوری را به پوشه example14 تغییر می‏دهیم و دستور زیر را اجرا می‏کنیم:

ionic setup sass

با این کار پیشنیازها را دانلود می‏کند، توضیحات را در فایل index.html پاک می‏کند و دو فایل ionic.app.css و ionic.app.min.css در داخل پوشه www/css ایجاد می‏کند. بسیار شیرین، درسته!

# کار با آیونیک scss

این بخش چگونگی شخصی سازی متغیرها و میکسین‏های scss را پوشش می‏دهد.

کدهایی را که نوشتیم فرض می‏کنیم که زمینه‏های لازم برای کار با SCSS را دارید.

اگر در scss تازه کار هستید به این راهنمایی‏ها توجه کنید:

<http://sass-lang.com/guide>.

## تم پایه

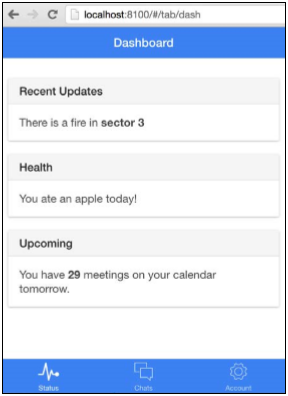
پیش از این، رنگ پایه قالب ارائه شده به وسیله آیونیک را دیدیم: مثبت، قاطع، آرام و ... . آن‏ها همه به وسیله تیم آیونیک از پیش تعریف و نصب شده‏اند. اگر شما می‏خواهید رنگ اجزا را به وسیله کلاس مثبت تغییر دهید؟ اجازه بدین به روند انجام این کار اشاره‏ای بکنیم.

با اشاره به پوشه example14، فایل www/index.html باز کنید و قسمت ion-nav-bar را از bar-stable to bar-positive به روز رسانی کنید. حالا، فایل www/ templates/tabs.html را باز کنید کلاسی با اسم tabs-color-active-positive را در مسیر ion-tabs پاک کنید. و tabs-positive. را به آن اضافه کنید.

برای دیدن خروجی دستور زیر را اجرا کنید:

ionic serve

رابط‌کاربری شبیه زیر می‏شود :



بیایید فرض کنیم این برنامه نهایی است. حالا میخواهیم یک تم جدید برای این صفحه بسازیم طراحی تم بسیار ساده است. همه رنگ‏های آبی را با سبز جایگزین می‏کنیم.

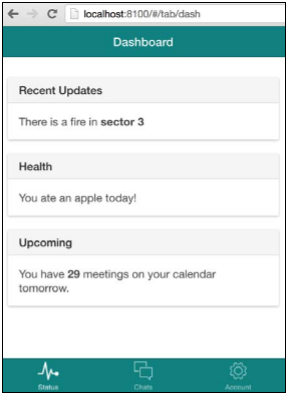
برای این کار فایل scss/ionic.app.scss را باز میکنیم و خط زیر را از قسمت کامنت‏ها کپی می‏کنیم:

$positive: #387ef5 !default;

و آن را بعد از کامنت‏ها(بعد از علامت /\*) قرار می‏دهیم. حالا، مقدار متغییر positive را به teal تغییر می‏دهیم:

$positive: teal;

تغییرات را ذخیره می‏کنیم و sass در پس زمینه اجرا میکنیم به صورت اتوماتیک فایل‏های جدید ionic.app.css و ionic.app.min.css تولید می‏شود. سپس صفحه به صورت اتوماتیک بارگذاری مجدد می‏شود. و شما صفحه را با تم جدید می‏بینید:



توجه کنید که تماتی رفرنس‏های کلاس positive به teal بروزرسانی شده‏اند. عالی نیست! مدیریت قالب برای برنامه‏های موبایل دیگر سخت نیست!

# درک راه‏اندازی آیونیک scss

در این بخش می‎بینیم که چطور scss نصب می‏شود.

اگر نگاهی قوانین پروژه ساخته شده بیاندازید، پوشه scss را می‏بینید، که شامل فایل ionic.app.scss است برای شخصی سازی قالب پیش‏فرض آیونیک، متغییرها را در اینجا باز نویسی کنید. البته فقط در اینجا تغییر متغییرها کافی است. اگر شما برای استفاده از چند قالب برنامه ریزی کرده‏اید، توصیه می‏کنم از فایل‏های theme1.scss و theme2.scss و ... در داخل پوشه scss استفاده کنید.

به یاد داشته باشید که هر فایل قالب باید دو خط زیر را دارا باشد:

// The path for our ionicons font files, relative to the built CSS

in www/css

$ionicons-font-path: "../lib/ionic/fonts" !default;

// Include all of Ionic

@import "www/lib/ionic/scss/ionic";

در یک فایل قالب معمولی، ابتدا متغییرهای scss را بازنویسی کنید. در بخش بعد فایل‏های هسته scss را بارگذاری کنید و در نهایت کلاس‏های تولید شده را باز نویسی کنید.

فایل‏های scss هسته آیونیک به دو خط زیر اشاره دارد:

$ionicons-font-path: "../lib/ionic/fonts" !default;

@import "www/lib/ionic/scss/ionic";

اگر شما قبلا با SCSS کارکرده باشید، شما تعریف یک متغییر را که با !defult اختصاص داده می‏شود یک مقدار جدید که...

برای درک بهتر آنچه که اتفاق می‏افتد، ابتدا مسیر متغییر ionicons-font-path$ را مشخص می‏کنیم. که در مسیر زیر است www/lib/ionic/fontsاین پوشه چهار فایل فونت را شامل می‏شود، که با توجه به سازگاری مرورگرها استفاده می‏شوند.

حال به محل نصب فریمورک SCSS می‏رویم. در مسیر زیر www/lib/ionic/scss/ionic . شما باید توجه داشته باشید، پوشه‏ای به نام ionic در داخل پوشه scss وجود ندارد. این بخاطر فایل ionic.scss در فولدر www/lib/ ionic/scss/ است. همچنین به دیگر فایل‏های SCSS داخل پوشه scss که با (\_) شروع می‏شوند توجه کنید.

اگر شما فایل ionic.scss را باز کنید، شما باید به همه فایل‏های ایمپورت شده SCSS در داخل فولدر جاری توجه کنید:

@charset "UTF-8";

@import

// Ionicons

"ionicons/ionicons.scss",

// Variables

"mixins",

"variables",

// Base

"reset",

"scaffolding",

"type",

// Components

"action-sheet",

"backdrop",

"bar",

"tabs",

"menu",

"modal",

"popover",

"popup",

"loading",

"items",

"list",

"badge",

"slide-box",

"refresher",

"spinner",

// Forms

"form",

"checkbox",

"toggle",

"radio",

"range",

"select",

"progress",

// Buttons

"button",

"button-bar",

// Util

"grid",

"util",

"platform",

// Animations

"animations",

"transitions";

اگر می‏خواهید هر تغییری در آیونیک بدهید، باید به فایل ionicons/ionicons. scss; مراجعه کنید یا برای تغییرات در در اجزا، باید به فایل \_modal.scss مراجعه کنید.اگر بخواهید تغییراتی در انیمیشن‏ها ایجاد کنید باید به فایل \_animations.scss مراجعه کنید.

دو تا فایل مهم وجود دارد که شما باید برای انجام تغییرات در اجزا آیونیک از آن مطلع باشید:

* \_variables.scss
* \_mixins.scss

به عنوان پیشنهادی برای نام‏ها، تمام متغیرها برای بازنویسی و میکسین‏ها برای استفاده مجدد به ترتیب ذخیره می‏شوند. اگر شما \_variables.scss را باز کنید ، متغییرها برا رنگ‏ها، فونت‏ها، پدینگ‏ها(حاشیه های داخلی)،مارجین‏ها(حاشیه‏های خارجی) و ... را می‏بینید.

به عنوان مثال، متن \_button را جستجو کنید، موجودیت‏هایی رو برای به وجود آوردن هر دکمه هستند را پیدا می‏کنید. با توجه به گزیده‏ای از این بخش:

$button-positive-bg: $positive !default;

$button-positive-text: #fff !default;

$button-positive-border: darken($positive, 10%) !default;

$button-positive-active-bg: darken($positive, 10%) !default;

$button-positive-active-border: darken($positive, 10%) !default;

می‏توانید ببینید که چگونه دکمه مورد نظر با استفاده از کلاس positive$ به وجود آمده است. با تغییر یک متغییر positive$ می‏بینید که دکمه به چه شکلی در می‏آید.

نمونه‏ای دیگر از بخش گرید:

// Grids

// ------------------------------

$grid-padding-width: 10px !default;

$grid-responsive-sm-break: 567px !default;

// smaller than landscape phone

$grid-responsive-md-break: 767px !default;

// smaller than portrait tablet

$grid-responsive-lg-break: 1023px !default;

// smaller than landscape tablet

اگر می‏خواهید تغییراتی در گرید خود ایجاد کنید، باید به قسمت بالا مراجعه کنید. بنابراین می‏توانید برای ابزار‏های کوچک، متوسط و بزرگ متغییر‏هایی را که در اینجا هستند تعریف کنید.

با صرف زمان برای این فایل و فهمیدن این که چه متغییرهایی را بازنویسی کرده‏اید.

# خلاصه

یادگرفتیم که برای کار چارچوب Ionic لازم است یک دید کلی از ساختار سکوی برنامه های ترکیبی موبایل داشته باشیم. مفهومی ساده و جالب دارد. تقریبا تمام سیستم عامل های موبایل (که هنگامی که با Cordova کار می‌کنیم سکو نامیده می شود) یک API[[21]](#footnote-21) برای توسعه برنامه‌هایش دارد. این API ها شامل مؤلفه‌ای است به نام **Web View**. این web view معمولاً یک مرورگر است که درون برنامه‌ی موبایل اجرا می‌‌شود. این مرورگر ها کدهای HTML,CSS را اجرا می کنند بدان معنی که شما صفحه را با استفاده از فناوری های وب توسعه و سپس آن را دورن برنامه خود اجرا می‌کنید.

# منابع

1. Build real-time and hybrid mobile applications with Ionic - Learning Ionic
2. مقالات سایت دات نت تیپس نوشته کاربر غلامرضا ربال <http://www.dotnettips.info/learningpaths/details/24>

1. Android [↑](#footnote-ref-1)
2. iOS [↑](#footnote-ref-2)
3. Windows Phone [↑](#footnote-ref-3)
4. ‌Black Berry [↑](#footnote-ref-4)
5. Framework یا چارچوب [↑](#footnote-ref-5)
6. hybrid [↑](#footnote-ref-6)
7. [ionicframework.com](http://www.ionicframework.com) [↑](#footnote-ref-7)
8. scaffold [↑](#footnote-ref-8)
9. Application Programming Interface [↑](#footnote-ref-9)
10. Framework [↑](#footnote-ref-10)
11. Software Development Kit [↑](#footnote-ref-11)
12. Front-end [↑](#footnote-ref-12)
13. Command Line Interface [↑](#footnote-ref-13)
14. Terminal [↑](#footnote-ref-14)
15. distributed version control [↑](#footnote-ref-15)
16. assets [↑](#footnote-ref-16)
17. production [↑](#footnote-ref-17)
18. Tab [↑](#footnote-ref-18)
19. navigation [↑](#footnote-ref-19)
20. Change Directory [↑](#footnote-ref-20)
21. Application Programming Interface [↑](#footnote-ref-21)