

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 همانم 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، همانم 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)

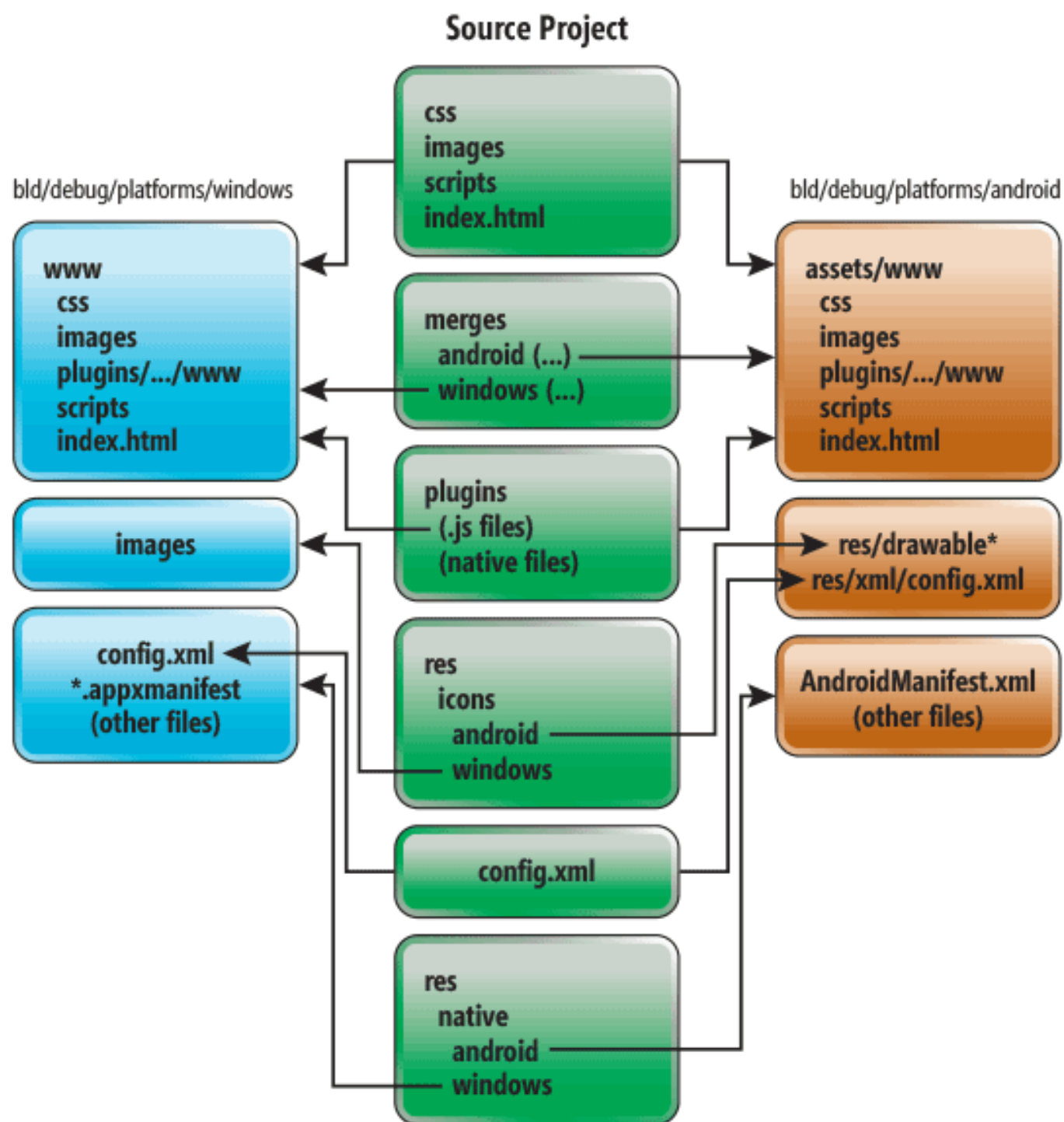
در زمان build، تمام فایل‌های موجود در merges/"platformname"، در فولدرهای هم نامی در شاخه‌ی ریشه‌ی پروژه کپی شده و جایگزین فایل‌های قبلی خواهند شد.

### 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 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);

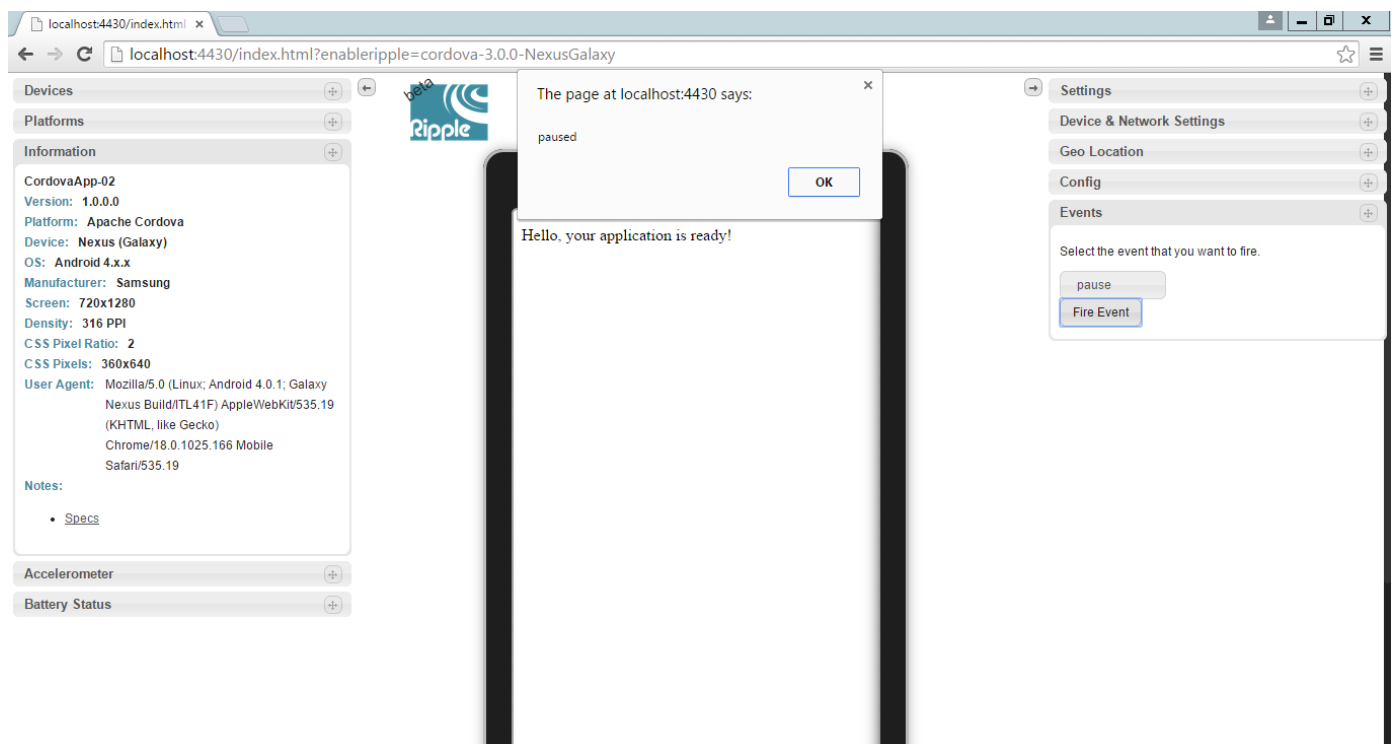
// TODO: Cordova has been loaded. Perform any initialization that requires Cordova here.
};

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 استودیو استفاده کنید:

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

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

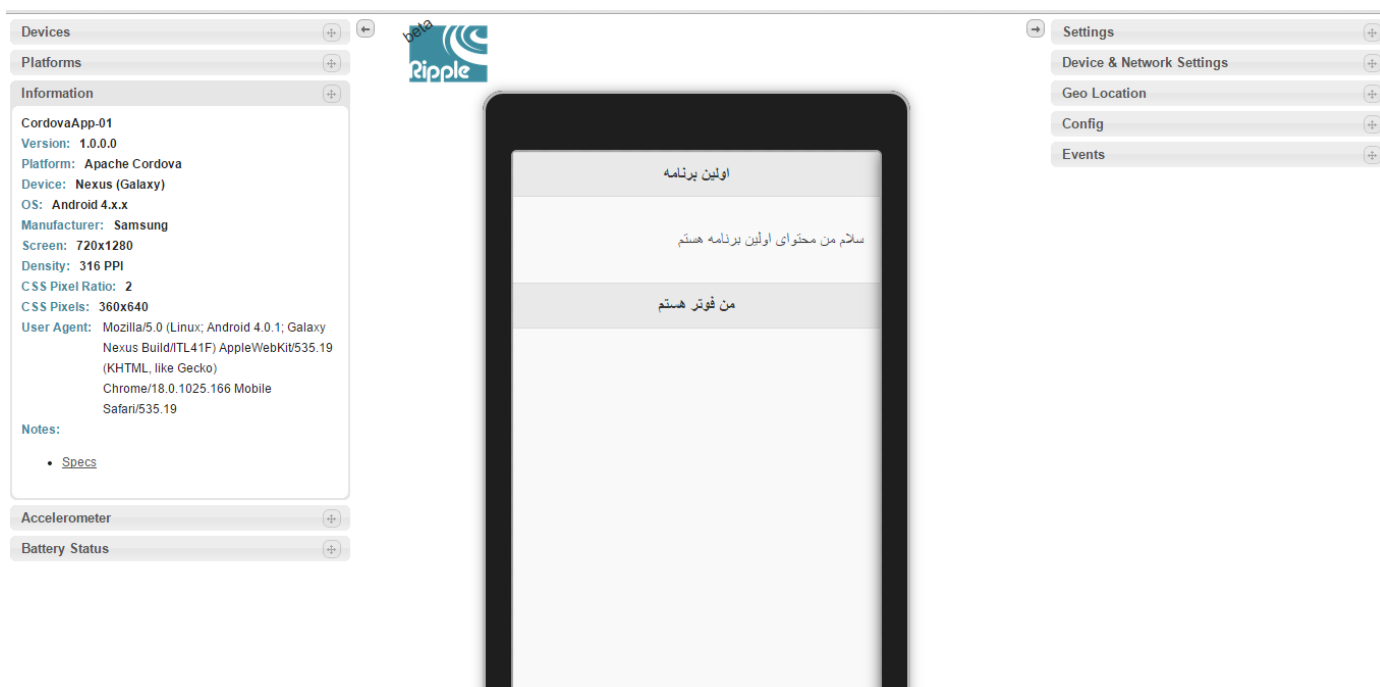
بعد از دانلود فایل‌های مورد نظر خود، فولدری بنام jquery.mobile.rtl در ریشه پروژه ایجاد خواهد شد. به ترتیب فایل‌های rtl.jquery.mobile-1.4.0.js و rtl.jquery.mobile-1.4.0.css موجود در زیر شاخه‌های فلدر مذکور را به head و آخر body فایل index.html اضافه کنید.

```
<!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.

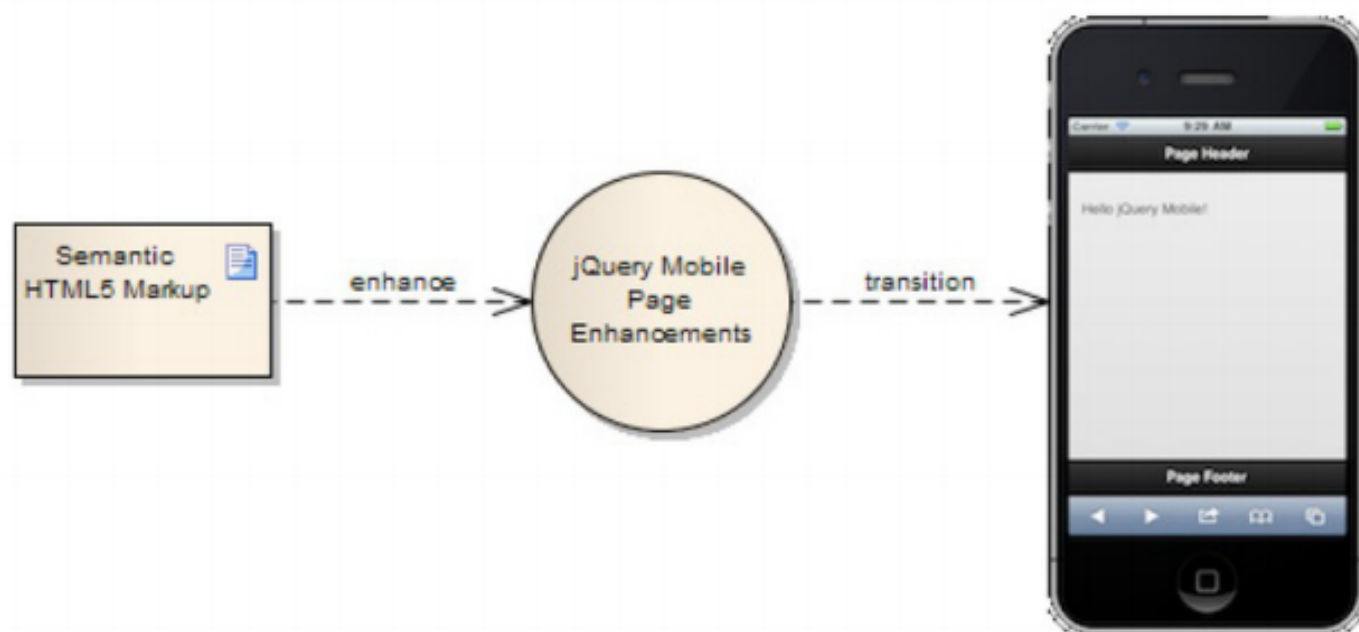
تگ متای زیر برای تنظیمات مربوط به viewport است و برای jQuery Mobile توصیه می‌شود.

```
<!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 ویجتی خاص، با آن آشنا خواهیم شد.

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

[دموی مربوط به ویجت‌ها](#)

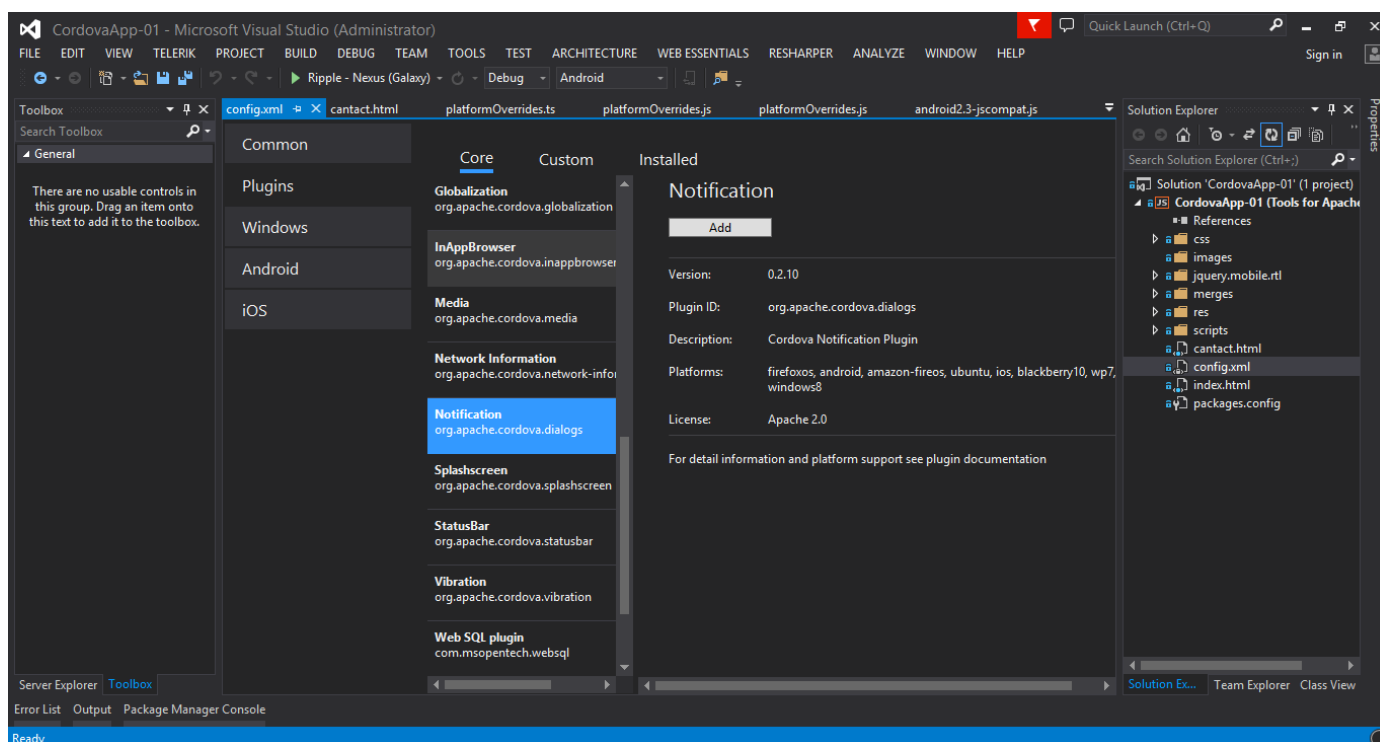
[لیست تمام رخدادهای](#)

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.

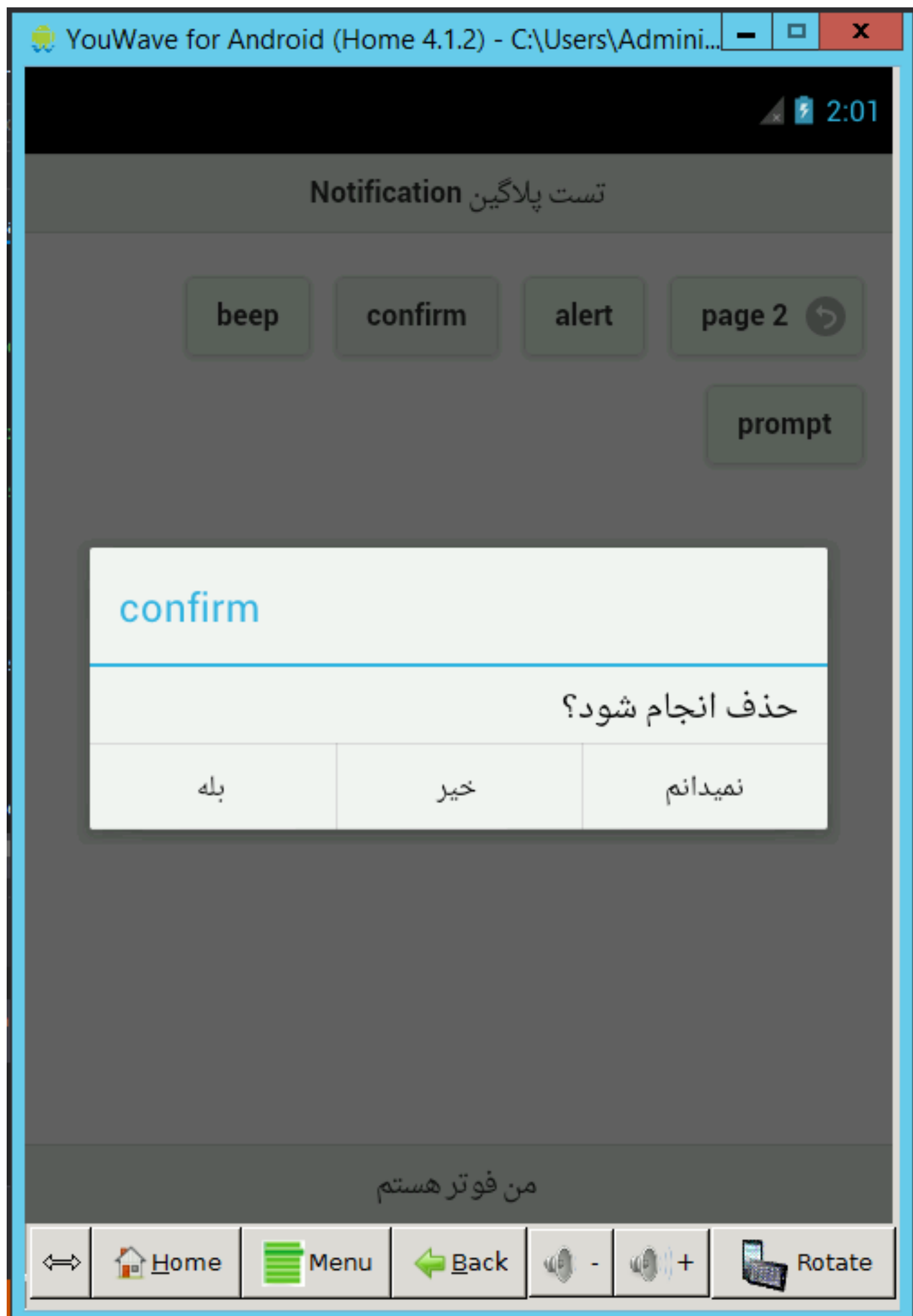
برای اینکار شبیه ساز [YouWave](#) را دانلود کرده و نصب کنید. سپس در قسمت toolbar ویژوال، گزینه‌ی Device را به جای شبیه ساز Ripple انتخاب کنید. نرم افزار youwave را اجرا کنید حال اگر برنامه را اجرا کنید با خطای زیر مواجه خواهید شد:

```
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.