



Thank you for purchasing revolution!

We tried to cover everything included in this package, if you need more clarification, feel free to contact us through Codecanyon contact form.

1. Cordova and Ionic

The package is based on Ionic framework (<http://ionicframework.com/>) and you should be familiar with techniques like HTML, CSS and AngularJS.

Setting up Ionic and dependencies is easy. Install Node.js on your computer (<https://nodejs.org/en/>) and in your terminal window (command on Windows), type:

```
npm install -g cordova ionic
```

This should setup Ionic and Cordova on your computer. You can learn more about this here: <http://ionicframework.com/getting-started/>

Now, if you navigate to app folder and type

```
ionic serve
```

you should be able to test the app on your browser.

To build the app for a platform, you should write:

```
ionic platform add ios  
$ ionic build ios
```

Replace ios with android or any other platform.

There are more to discuss, but you can learn more about cordova and Ionic following the links:

<http://ionicframework.com/getting-started/>
<http://ionicframework.com/docs/>
<https://cordova.apache.org/docs/en/latest/>
<https://cordova.apache.org/plugins/>

2. Package elements

We have included in the package elements to help you start your new app with ease.

- Design elements
 - Header menu with effects
 - Side menus - custom made
 - Notifications (right side menu) with options (swipe functionality)
 - Slider
 - Banner and icon sections
 - Owl carousel
 - Tabs - custom made
 - Custom form and special elements
 - Footer
 - SCSS way
 - Many color combinations
- Functionalities
 - Google Feed API (Angular style)
 - Wordpress Feed (Angular again)
 - Firebase (AngularFire) simple stuff
 - Plugins
 - Toast message
 - Rate my app dialog
 - Barcode scanner
 - Device info

That's it so far. But this is just the beginning. We will continue updating this

template and add a lot more features and design elements.

3. Design elements

We have styled Ionic elements to fit this design, of course you can add your own CSS code if you want.

And we added a `www/css/helper.css` file, is a CSS framework created by us, every class is intuitive (
`.p30` stands for `padding:30px`;
`.mt10` equals `margin-top: 10px`;

and so on, just check `www/css/helpers.css` file to have a better sense.

Every template for each screen you can find inside `www/templates` folder. Just include Ionic elements you want to use. Follow this link to see every Ionic element: <http://ionicframework.com/docs/components/>.

4. Functionalities

Everything you need you can find in `www/js` folder.

app.js

Here is where the router is defined. Pages and controllers definitions.

```
.state('app.plugins', {  
  url: '/plugins',  
  views: {  
    'menuContent': {  
      templateUrl: 'templates/plugins.html',  
      controller: 'PluginsCtrl'  
    }  
  }  
})
```

This is how the plugins page is added. `templateUrl` is where the physical

page is saved (www/templates folder) and controller is the controller that the page will use when accessed. This way, you can add a new page to the template:

add this to your www/js/app.js file:

```
.state('app.custom', {  
  url: '/custom',  
  views: {  
    'menuContent': {  
      templateUrl: 'templates/custom.html',  
      controller: 'CustomCtrl'  
    }  
  }  
})
```

and create a new page inside www/templates folder, and save it as custom.html.

and of course, inside www/js/controllers.js create a new controller:

```
app.controller('CustomCtrl', function($scope) {  
  // your functions here  
})
```

Add a new link to the page, maybe in your left side menu:

open www/templates/menu.html and add this to the items list:

```
<ion-item menu-close href="#/app/custom" class="light border-b-xs">  
  Custom  
</ion-item>
```

Now, you should be able to tap on the Custom label menu and navigate to your page.

controllers.js

Here is where your functions are defined.

AppCtrl

This is where the menus/main functions are defined.

We define here the function to change the navigation icons (from bars to X) when the side menu is opened.

Next we have the right menu notifications functionality and data defined.

UserCtrl

Here is a standard redirect to the home screen when tap on login button. There is no user management included at this point, but we are working on this add-on and it will be released soon. We will use Firebase for that.

HomeCtrl

Here are the home screen functions. Top menu animation effect and tabs are defined here.

NewsCtrl

Here we arrange the data we get from the news service (we will talk about it later) and store it in a \$scope to be displayed in the page. And we call the function like this: `getNews(10);`. If you want to display less articles, you can replace 10 in this statement.

BlogCtrl

Here we arrange the data we get from the blog service (we will talk about it later) and store it in a \$scope to be displayed in the page. We get Wordpress categories, posts and individual post content.

FirebaseCtrl

Here we included a few things to have a start with your Firebase project. There is a 3-way data binding sample, to read and update a Firebase object realtime, and there is a simple chat, illustrating the use of 3-way data binding using arrays of data.

The functions are AngularJS way, AngularFire more precisely, and we will come back to this section to add a lot more functionalities.

ElementsCtrl

As you will see, this controller is empty, here we don't have any functionality since there are design elements only.

PluginsCtrl

Here we have some NgCordova plugins, to illustrate the way you can include device capable functions on your app. Note that the plugins will work only when testing on a real device.

Toast plugin displays a small message on the screen: read more <http://ngcordova.com/docs/plugins/toast/>

Rate my app plugin displays a popup to rate the app: read more <http://ngcordova.com/docs/plugins/appRate/>

Barcode scanner plugin allows you to scan any barcode and display useful info: read more <http://ngcordova.com/docs/plugins/barcodeScanner/>

Device info is a useful plugin to get device info: read more <http://ngcordova.com/docs/plugins/device/>

firebase.init

This initialize Firebase with your credentials.

Create a Firebase account (if you don't have one already): <https://firebase.google.com/>

Go to Firebase Console and create a new project. Click on Add Firebase to your web app and include your credentials from here in `firebase.init` page.

If you want to learn more about AngularFire functions, feel free to read this doc: <https://github.com/firebase/angularfire/blob/master/docs/quickstart.md>
Since your app is based on AngularJS, be sure to include Angular functions only.

services.js

Here we define most of the data sources.

FeedLoader

This factory set up the Google feed api link. Is standard, so you don't have to change it.

FeedList

This service is where we setup the news sources. We added `time.com`, `mashable` and `wordpress` for now, but you can add as many as you want, using the model:

```
var feedSources = [  
  {title: 'Time', url: 'http://time.com/feed/'},  
  {title: 'Mashable', url: 'http://mashable.com/feed/'},  
  {title: 'Wordpress', url: 'https://en.blog.wordpress.com/feed/'},  
  {title: 'My custom feed', url: 'https://mycustomfeed.com/feed/'}  
];
```

Blog

This service is where we setup the blog(Wordpress) feed.
Just include your wordpress address url like this:

```
var address = 'http://mywordpresswebsite.com/wp/api/';
```

Firebase

This service is where we setup the Firebase url. Add your own Firebase app url like this:

```
this.url = function(id) {  
    var url = "https://myfirebaseurl.firebaseio.com";  
    return url;  
};
```

Owl Carousel

Here is where we setup Owl carousel to work with AngularJS. You don't have anything to setup here.

That's it for now, we will revise this documentation with every new release.

Have fun, and don't forget to get in touch if you're having troubles.

