Rubyonic

Mobile Ui Pack for IonicFramework

Introduction

Thank you for buying rubyonic!

Color Scheme Manager

SASS Structure & In-code Comments

New! Tinder Card-Swipe Functionality!

Preamble

- A working Knowledge of IonicFramework is needed in setting up rubyonic. If you need help in setting up ionic for the first time visit the Getting Started Guide
- A working knowledge of SASS is also needed. You can read the SASS Guide for Beginners and Ionic SASS tutorial for more information

Installation

Clean setup

1. Setup a blank ionic project (SASS must be setup as well):
   `ionic start myApp blank --sass`
2. Unzip the rubyonic.zip file. You'll have a **scss** folder and **www** folder
3. Replace **<ionicproject>/scss** with the unzipped **scss** and **<ionicproject>/www** with the unzipped **www** folder

Existing Installation

It is recommended you use a use rubyonic on a clean setup to understand it better

1. Unzip the rubyonic.zip file. You'll have a **scss** folder and **www** folder

2. Ensure your app already has sass setup. If not, run the following command:

```
ionic setup sass
```

3. Open **rubyonic.zip/scss/ionic.app.scss** and **note the code (//Rubyonic Files). You'll add this to your own <ionicproject>ionic.app.scss**

```
21  // Rubyonic Files
22  @import 'rubyonic/partials/colors';
23  @import 'rubyonic/partials/typography';
24
25  // Include all of Ionic
26  @import "www/lib/ionic/scss/ionic";
27
28  // Rubyonic Files
29  @import 'rubyonic/partials/mixins';
30  @import 'rubyonic/partials/theme';
31  @import 'rubyonic/partials/buttons';
32  @import 'rubyonic/partials/general';
33  @import 'rubyonic/screens/login';
34  @import 'rubyonic/screens/feed';
35  @import 'rubyonic/screens/sidebar';
36  @import 'rubyonic/screens/chat';
37  @import 'rubyonic/screens/profile';
38  @import 'rubyonic/screens/chat-ui';
39  @import 'rubyonic/screens/viewpost';
40  @import 'rubyonic/partials/forms';
41
```

4. Copy the **SCSS/rubyonic** folder into your **<project>/scss folder**
5. Open up your **<ionicproject>/scss/ionic.app.scss** and add code from (3)

Template Files and assets

1. Copy the **fonts**, **img**, and **templates/rubyonic** folder to your **<ionicproject>/www** folder
2. Access **rubyonic.zip/www/js/app.js** and copy the defined states into your own app.js (or file holding your routes)

File Structure

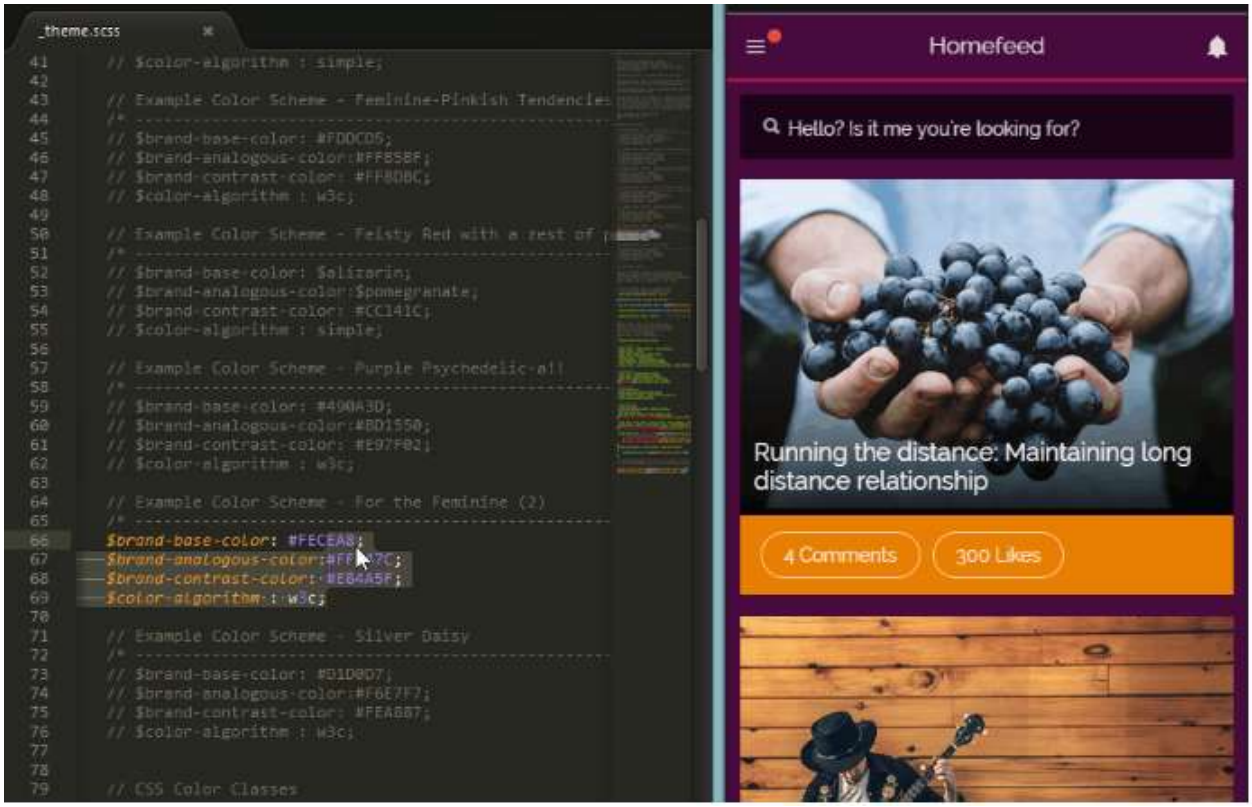
```
www-------------
       img
       fonts (Raleway & Ubuntu are included. refer to
       rubyonic/scss/partials/typography.scss for more info

       js     (open the app.js to see the routes)
       templates
```

```
scss-------------
       rubyonic
              screens
              partials
```

Color Scheme Manager

Changing the color scheme of the app is easy, but we need to understand what happens behind the scenes.



Changing Rubyonic's color scheme involves changing the following four variables in rubyonic's **theme.scss** file:

- **$brand-base-color**: The Main Color for the app
- **$brand-analogous-color**: The second main color for the app. use for subtle contrast
- **$brand-contrast-color**: A major contrast color to either
- **Color Algorithm: (For calculating the best text color contrast over $brand-base-color**

Color Algorithm tackles another issue with changing colors: **text contrast.** This ensures that the text will always have the best matching contrast for any of the Main/Analogous colors. That means you

don't have to worry if the text doesn't stand out.

Rubyonic has two methods to calculate this: **simple** and **w3c** . switching between these should fix any text-color issues that may come up. We've also included a set of color matching schemes you can use right away.

Extending the color scheme to new templates

If you need to customize an element's text color to always contrast the main brand color, add it to the set of existing rules in your **<ionicproject>/scss/rubyonic/_theme.scss**. As **_theme.scss** has rules from all the other scss files Its good to group rules with a comment so you'll always know where a rule should go if you need move/overide it. All the rules below have one style.

```
Let the comments guide you to your destination :D
*/
.button.brand-base-text-color,

// sidebar.scss
.menu-screen .item-complex .item-content,
.menu-screen .item,
.item-note.brand-base-text-color,
.menu-screen .group-header,
.notifications .notification-title,
.notifications .notification-summary,
.menu-screen .list.menu-screen-searchbox .item.item-input .placeholder-icon,

// chat.scss
.chat-list .notification-title,
.chat-list .notification-summary,
.tabs.chat-footer-controls .tab-item,
ion-footer-bar.chat-footer-bar .button,

// feed.scss
.feed-posted-by,
.feed-time-post.brand-base-color,
.feed-item.is-quote, .feed-item.is-quote *,
.feed-item.is-quote:before,

// chat-ui.scss
.chat-message,
.chat-bar-contact-name .contact-title,
.chat-text-input,
.bar.bar-chat .button-clear.button-,

// form.scss
.input-label
{@include text-color-contrast($brand-base-color, $color-algorithm);}
```

Update on Color Manager

We've got feedback on issues using the color manager from the get go following these instructions. The issue comes from the manner node/grunt version execute. If your color scheme isn't changing as expected, do the following:

- Try deleting your **existing /node_modules** directory and running **npm update** while in <**project folder**>
- update your node version.

New! Tinder Screens!

Tinder screens have been added. They are based on the ionic-ion-tinder-cards with updates currently on the way. It also features the same color customization as the other rubyonic templates.

Features

- 4 different tinder styles based off ionic-tinder cards (and improved)
- swyped-card ready functions - exposed data per card for data collection/saving operations
- uses the rubyonic color manager

Installation

Copy the files to their destinations listed below.

Files	Location in Product File	Destination on your <ionic project>
Tinder images	Rubyonic.zip/www/Img/demo	<yourIonicProject>/www/img/demo
Tinder.scss	Rubyonic.zip/scss/rubyonic/screens/	<yourIonicProject>/scss/rubyonic/screens/
Apps.js, collide.js,controllers.js, ionic.tdcards.js	Rubyonic.zip/www/js (Apps.js houses the added swiping functionality. if you're using an existing installation skip app.js and head to the next section	<yourIonicProject>/www/js/

Setup

Setup your file references in your <ionic-project>/www/index.html like so:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, w
    <title></title>


    <!-- compiled css output -->
    <link href="css/ionic.app.css" rel="stylesheet">

    <!-- ionic/angularjs js -->
    <script src="lib/ionic/js/ionic.bundle.js"></script>
    <script src="js/collide.js"></script>
    <script src="js/ionic.tdcards.js"></script>

    <!-- cordova script (this will be a 404 during development) -->
    <script src="cordova.js"></script>

    <!-- your app's js -->
    <script src="js/app.js"></script>
    <script src="js/controllers.js"></script>
  </head>

  <body ng-app="starter">
    <ion-nav-view></ion-nav-view>
  </body>
</html>

```

Ensure you have your dependencies added to your angular module:

```
// angular.module is a global place for creating, registering and retrieving Angular modules
// 'starter' is the name of this angular module example (also set in a <body> attribute in i
// the 2nd parameter is an array of 'requires'
// 'starter.controllers' is found in controllers.js
angular.module('starter', ['ionic', 'starter.controllers', 'ionic.contrib.ui.tinderCards'])

.run(function($ionicPlatform) {
  $ionicPlatform.ready(function() {
    // Hide the accessory bar by default (remove this to show the accessory bar above the ke
    // for form inputs)
    ionic.Platform.fullScreen()
    if (window.cordova && window.cordova.plugins.Keyboard) {
      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
    }
    if (window.StatusBar) {
      // org.apache.cordova.statusbar required
      // StatusBar.styleDefault();
      StatusBar.hide();
    }
    StatusBar.hide();
  });
})

.config(function($stateProvider, $urlRouterProvider) {
  $stateProvider

  .state('app', {
    url: "/app",
```

Copy the routes to your app

```
  .state('app.tinder-one', {
    url: "/tinder-one",
    views: {
      'menuContent': {
        templateUrl: "templates/tinder/tinder-one.html"
      }
    }
  })

  .state('app.tinder-two', {
    url: "/tinder-two",
    views: {
      'menuContent': {
        templateUrl: "templates/tinder/tinder-two.html"
      }
    }
  })

  .state('app.tinder-three', {
    url: "/tinder-three",
    views: {
      'menuContent': {
        templateUrl: "templates/tinder/tinder-three.html"
      }
    }
  })

  .state('app.tinder-four', {
    url: "/tinder-four",
```

Add the reference to tinder.scss to your ionic.app.scss

```
$assertive:                     #ef473a !default;
$royal:                         #886aea !default;
$dark:                          #444 !default;
*/

// The path for our ionicons font files, relative to the built CSS in www/css
$ionicons-font-path: "../lib/ionic/fonts" !default;

// Rubyonic Files
@import 'rubyonic/partials/colors';
@import 'rubyonic/partials/typography';

// Include all of Ionic
@import "www/lib/ionic/scss/ionic";

// Rubyonic Files
@import 'rubyonic/partials/mixins';
@import 'rubyonic/partials/theme';
@import 'rubyonic/partials/buttons';
@import 'rubyonic/partials/general';
@import 'rubyonic/screens/login';
@import 'rubyonic/screens/feed';
@import 'rubyonic/screens/sidebar';
@import 'rubyonic/screens/chat';
@import 'rubyonic/screens/profile';
@import 'rubyonic/screens/chat-ui';
@import 'rubyonic/screens/viewpost';
@import 'rubyonic/partials/forms';
@import 'rubyonic/screens/tinder';
```

Lastly, add the Controller code for tinder card functionality:

```
})


.controller('JCardsCtrl', function($scope) {
  console.log('CARDS CTRL Initiated');
  var cardTypes = [
    {
      image: 'img/demo/tinder-full-pic.jpg',
      title: 'Samantha Gamblesx', location: 'Manchester',
      description: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit.',
      vipStatus : false
    },
    {
      image: 'img/demo/tinder-full-pic-2.jpg',
      title: 'Junior Max', location: 'Manchester',
      description: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit.',
      vipStatus : true
    },
    {
      image: 'img/demo/tinder-full-pic-3.jpg',
      title: 'Karla Valentine', location: 'Manchester',
      description: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit.',
      vipStatus : true
    }
  ];

  $scope.cardsControl = {};

  $scope.reload = function() {
```

And you're done! access the cards using the routes you defined via the command prompt

```
ionic serve
goto <#/app/tinder-two>
```

(we're using the default setup of routes with the ionic starter template. Yours may be different)

For support and feedback, send us a mail at contact@audacitus.com

Regards,