

The Jackal of Javascript

Knowing is different & Doing is different



HOME ABOUT ME HISTORY OF THE WEB JAVASCRIPT 101 REQUEST

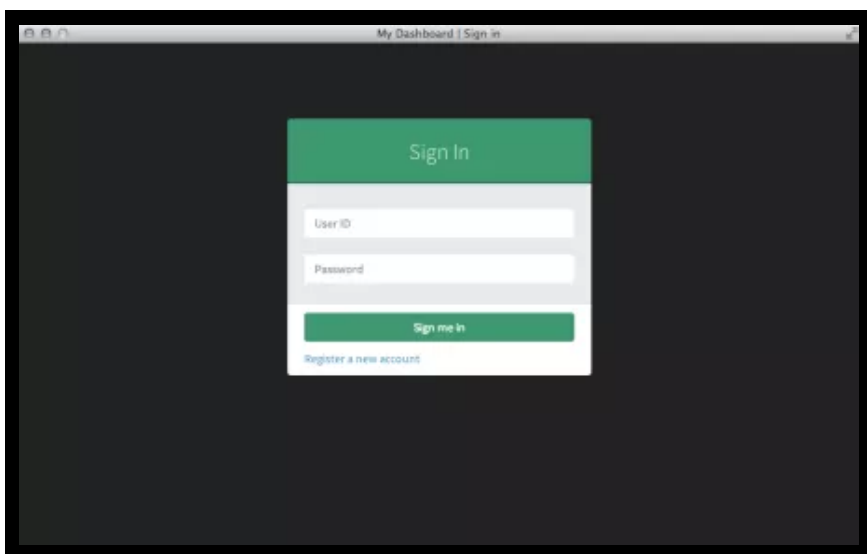


Node Webkit powered Dashboard application

This post is part 2 of 7 in the series [Node Webkit Apps](#)



In this post, we will be building a full fledged desktop application using node webkit and web technologies. We are going to build a dashboard app called *myDashboard*. The app will show the Weather report, Google Map, Flickr images and a Calendar. The final product would look like



This is me!!



Hello.. I am Arvind Ravulavaru a Full Stack Consultant based out of Hyderabad, India. This blog is my way of giving back to the Javascript Community!

[Know more about me](#)

Follow @arvindr21

[Follow @arvindr21](#)

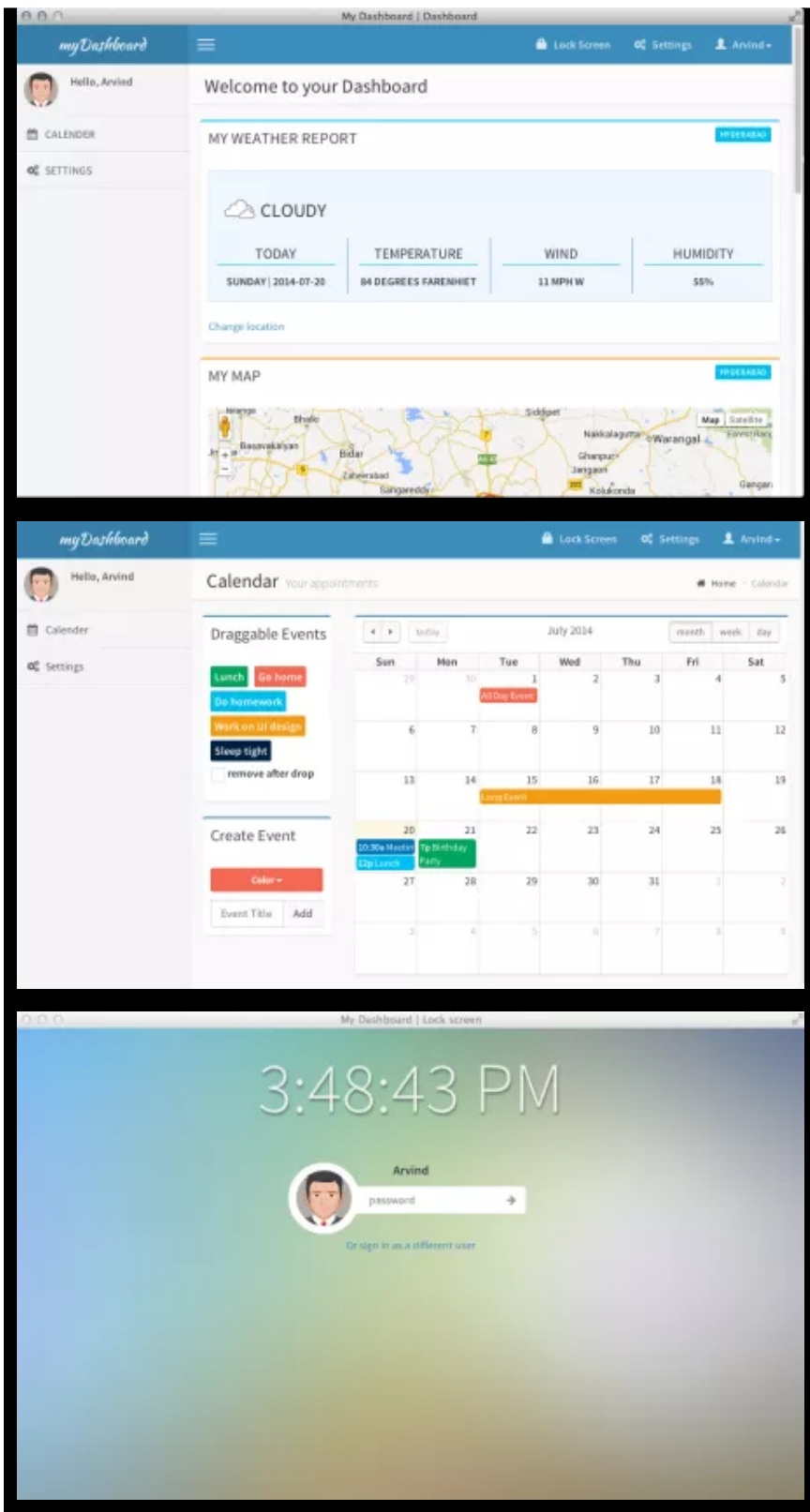
[Linked In](#)

[Google Plus](#)

[Mongo Director](#)

[Site Point](#)

Chronicles of the Jackal of Javascript



Chrome App

Chrome Extension

Workshops

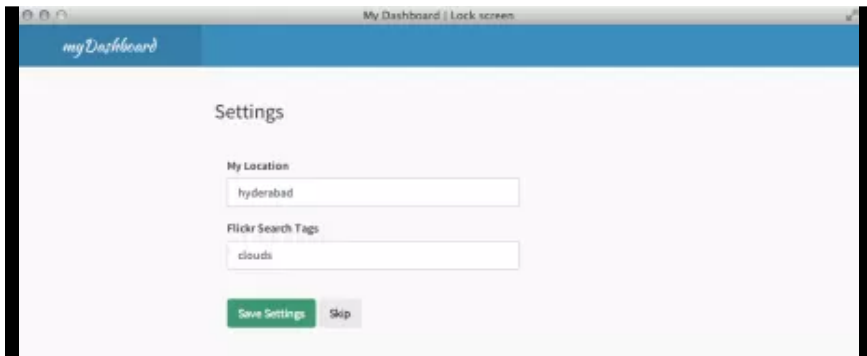
- \o/ M.E.A.N. Workshop

Presentations

- Overview of Angularjs
- How Browser Works
- Web Components
- Modern UI Build Systems and Scaffolding Tools
- Backbonejs
- Responsive Web Design

Projects

- Rotten Tomatoes jQuery Rest Client
- blueimp-file-upload-expressjs - A Node Package
- lazyboy - Command line laziness redefined
- diskDB - A Light Weight Disk based JSON Database - A Node Package
- My Slush Generators
- My Yeoman Generators



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



- My Web Components
- My Bower Packages

Series

Node Scheduler

– A Reminder App

- Node Webkit Apps
- Raspberry Pi

You can get the complete code [here](#). The final installers are around 200MB, hence did not share the installer.

So, let us see how we can build this app.

Contents

- Prerequisites
- Getting Started
- Begin Development
- Continue Development
- Distributing the application

Prerequisites

To build desktop apps, you need to have a general awareness on how web apps need to behave when on a desktop. This includes the menus, toolbars etc. You need to be also aware of the fact that internet is not always present and there might permissions issues to run the application.

You also need to have a good understanding of web technologies (HTML, CSS and JS), REST API, oAuth etc. Node webkit uses the power of Node js and Webkit engine to provide a platform to run Web apps driven by Nodejs on a desktop.

Before you continue, please go through [Hello Node](#) and [Node Webkit – Build Desktop Apps with Node and Web](#)

Guest Posts


- Building a Chat App with node-webkit, Firebase, and AngularJS
- Creating a Sentiment Analysis Application Using Node.js
- Implementing pagination with MongoDB, Express.js & Slush
- Creating a Firebase Powered End to End Ionic Application
- Getting started with MongoDB and Mongoose
- Yeoman, Mongoose and MongoDB

Top5 Posts

Technologies.

Getting Started

As mentioned earlier, we are going to build a Dashboard application. The first thing we are going to do is analyse what we are going to show in the dashboard.

Learn your way around Ionic framework. Check out Learning Ionic! [Buy](#) 

So, for this dashboard app, we are going to have a login and a register screen. We will also implement a lock screen for our dashboard app. On the main page, we are going to show a Weather report, a Google map and a few images from Flickr. Apart from these, we will also show a calendar.

We will let the user configure the location for the Weather as well the location where the map will be centered. We will also let the user provide the tags with which the Flickr search is made.

Now, we have an idea what we are going to do. It is up to you, if you want to design your application first or if you are lazy like me you will be searching for a Admin Dashboard template.


After a quick search. I found [AdminLTE Dasboard and Control Panel Template](#) which in my opinion is one of the best Free Admin templates out there. Kudos to [Abdullah Almsaeed](#) for open sourcing the template.

← We will be consuming this template and optimizing a couple of things for desktop apps. →

Begin Development

First, create a new folder named *myDashboard*. Next, create a new folder inside it named *template*. We will be dumping the template here and pick only the required components.

Download the [template](#) into the *template* folder and unzip it.

- [Architecting a Secure RESTful Node.js app](#) - 161,811 views
- [Getting Started with Angular 2.0](#)  235 views
- [Building a desktop app with Angular 2.0](#) - 86,982 views
- [Node Webkit – Build Desktop Apps with Node and Web Technologies](#) - 85,002 views
- [Ionic Restify MongoDB – An End to End Hybrid App](#) - 67,986 views

Recent Posts

- [Learning Ionic](#)
- [Electron, WordPress & Angular Material – An Offline Viewer](#)
- [Developing a MEAN app with Angular 2.0](#)
- [Pushbots and Cordova – Easy Push Notifications for your app](#) →
- [Getting Started with Client Side Storage](#)

Categories

- [Architecture](#) (8)
- [CSS](#) (9)
 - [CSS3](#) (6)

Next, inside *myDashboard* folder, **download the node-webkit libs** and unzip it. Rename the unzipped folder to *nw* and get rid of the node-webkit zip file.

Finally, create a new folder named *development*. This is where we do all the development.

Learn your way around Ionic framework. Check out Learning Ionic! [Buy](#)



Now, open a new terminal/prompt inside the development folder. We will initialize a new node project. Run

```
npm init
```

and fill it up like



This is the default node specific *package.json*. Now, we will be adding the dashboard app related configs. The final *package.json* would look like

```
package.json JavaScript
1 {
2   "name": "myDashboard",
3   "version": "0.0.1",
4   "description": "A node-webkit dashboard application",
5   "main": "index.html",
6   "private": true,
7   "keywords": [
8     "node",
9     "webkit",
10    "dashbaord",
11    "win",
12    "osx"
13  ],
14  "window": {
15    "toolbar": false,
16    "resizable": false,
17    "width": 1000,
18    "height": 600,
19    "min_width": 600,
20    "min_height": 400,
21    "max_width": 1000,
22    "max_height": 600,
23    "position": "center"
24  },
25  "author": "Arvind Ravulavaru",
26  "license": "MIT"
27 }
```

Now, inside the *development* folder, create a new file named *index.html*. This will be our starter file and fill it up like

```
index.html
```

- LESS (3)
- SASS (1)
- Database (17)
 - Disk DB (6)
 - Firebas
 - Mongol
 - MongoDB (3)
- Deployments (4)
 - Heroku (2)
- Desktop Apps (8)
- DNA Analysis (1)
- ECMAScript 6 (3)
- Electron (atom shell) (1)
- Frameworks (42)
 - Angular 2.0 (2)
 - Angularjs (21)
 - Backbonejs (1)
 - Bootstrap (9)
 - Emberjs (1)
 - Foundation (3)
 - Framework7 (1)
 - Ionic (11)
 - Lumx (1)
 - MEAN (4)
 - Meteorjs (1)
 - Onsen UI (2)
- Git (12)
 - GitHub (10)
- HTML (22)
 - HTML5 (21)
 - WebRTC (1)
 - WebSockets (6)
- Intel's App UI (1)
- IoT (6)
- Javascript (75)
 - jQuery (14)
- Material Design (3)
- Mobile (21)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Dashboard</title>
5 </head>
6 <body>
7   <h1>Hello World!</h1>
8 </body>
9 </html>

```

Now, we will run the app. We have 2 ways of running the

~~~

Learn your way around Ionic framework. Check out Learning Ionic! [Buy](#)



First way is to zip all the *contents* of *development* folder and launching it using the *nw* executable we have downloaded earlier. This is a very very very tedious process when the app starts to grow.

Second way is to configure sublime text to build the app for us (*recommended*). Check [Debugging with Sublime Text 2 and 3](#) for more info.

Once this is configured, open *index.html* and navigate to tools and select *node-webkit* as the build system




Now, to run the app *tools > build* or *Ctrl + B* or *Command + B* and you should see the app launch and the *Hello World* message.

During development you can set `"toolbar": true` to access the dev tools.

## ← Continue Development →

Now, we have set up the basic project. Let us add the required features. Before we do that, I would highly recommend going through the entire template again. This will give you an idea as what all features are already implemented and available to be directly consumed.

First up, we are going to implement is the registration and login feature. Before that, we will give a structure to the app. Inside *development* folder, create a new folder named *assets*. Inside *assets* create 4 folders named *css*, *js*, *img*, *fonts*.

- [Android](#) (15)
- [Firefox OS](#) (2)
- [iOS](#) (11)
- [PhoneGap](#) (16)
- [MQTT](#) (1)
- [My Books](#) (  )
- [Node](#) (60)
- [ExpressJS](#) (18)
- [Genome](#) (1)
- [Grunt](#) (8)
- [Koa](#) (1)
- [Node Webkit](#) (7)
- [NPM](#) (15)
- [Restify](#) (3)
- [Push Notifications](#) (1)
- [Quick and Dirty](#) (42)
- [Raspberry Pi](#) (9)
- [Setup](#) (60)
- [Testing](#) (5)
  - [Jasmine](#) (1)
  - [Protractor](#) (1)
  - [Test Automation](#) (1)
  - [Test Driven Development](#) (2)
- [Third Party Products](#) (4)
  - [Twilio](#) (4)
- [Tutorial](#) (79)
- [UI Performance](#) (1)
- [Web Components](#) (3)
- [Web Starter Kit](#) (1)

## Archives

- [July 2015](#)
- [May 2015](#)
- [April 2015](#)

Since I have already implemented the app, I know what all files we would be needing. So we will directly copy the files needed for the complete project.

### JS folder : Copy

- *myDashboard/template/AdminLTE-master/js/jquery-ui-1.10.3.min.js* to *myDashboard/development/assets/js/*
- *myDashboard/template/AdminLTE-master/js/AdminLTE/app.js* to *myDashboard/development/assets/js/*
- *myDashboard/template/AdminLTE-master/js/plugins/fullcalendar/fullcalendar.min.js* to *myDashboard/development/assets/js/*
- Download a copy of <http://ajax.googleapis.com/ajax/libs/jquery/2.0.2/jquery.min.js> to *myDashboard/development/assets/js/*

- March 2015
- February 2015
- January 2015
- December 2014
- November
- October 2014 
- September 2014
- July 2014
- June 2014
- May 2014
- April 2014
- March 2014
- February 2014


[Buy](#)

Inform me about new posts

Email \*

SUBSCRIBE!

### CSS folder : Copy

- *myDashboard/template/AdminLTE-master/css/AdminLTE.css* to *myDashboard/development/assets/css/*
- *myDashboard/template/AdminLTE-master/css/bootstrap.min.css* to *myDashboard/development/assets/css/*
- *myDashboard/template/AdminLTE-master/css/font-awesome.min.css* to *myDashboard/development/assets/css/*
- *myDashboard/template/AdminLTE-master/css/ionicons.min.css* to *myDashboard/development/assets/css/*
- *myDashboard/template/AdminLTE-master/css/fullcalendar/fullcalendar.css* to *myDashboard/development/assets/css/*



### Images folder : Copy

- *myDashboard/template/AdminLTE-master/img/avatar5.png* to *myDashboard/development/assets/img/*



- *myDashboard/template/AdminLTE-master/img/blur-background09.jpg* to *myDashboard/development/assets/img/*

## Fonts folder : Copy



- *myDashboard/template/AdminLTE-master/fonts* to *myDashboard/development/assets/fonts/*

Learn your way around Ionic framework. Check out Learning Ionic!

Buy



Once this tedious task is done, you need not worry about resources any more.

*index.html* will contain the login screen. You can refer *myDashboard/template/AdminLTE-master/pages/examples/login.html* for the login page. I have taken then above page, removed a few imports and fields and made it like

```
index.html
1  <!DOCTYPE html>
2  <html class="bg-black">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My Dashboard | Sign in</title>
7      <meta content="width=device-width, initial-scale=1, maximum
8      <!-- bootstrap 3.0.2 -->
9      <link href="assets/css/bootstrap.min.css" rel="stylesheet"
10     <!-- font Awesome -->
11     <link href="assets/css/font-awesome.min.css" rel="styleshee
12     <!-- Theme style -->
13     <link href="assets/css/AdminLTE.css" rel="stylesheet" type=
14 </head>
15
16 <body class="bg-black">
17
18     <div class="form-box" id="login-box">
19         <div class="header">Sign In</div>
20         <form action="dashboard.html" method="post">
21             <div class="body bg-gray">
22                 <div class="form-group has-error">
23                     <label class="control-label hide" id="form-
24                 </div>
25                 <div class="form-group has-error">
26                     <label class="control-label hide" for="user
27                     <input type="text" name="userid" id="userid
28                 </div>
29                 <div class="form-group has-error">
30                     <label class="control-label hide" for="pass
31                     <input type="password" name="password" id="
32                 </div>
33             </div>
34             <div class="footer">
35                 <button type="button" id="login" class="btn bg-
36                 <a href="register.html" class="text-center">Reg
37             </div>
38         </form>
39     </div>
40
41     <!-- jQuery 2.0.2 -->
42     <script src="assets/js/jquery.min.js"></script>
43     <script type="text/javascript">
```



```

44     $('#userid').on('blur', function() {
45         var $this = $(this);
46         if ($this.val().trim().length < 1) {
47             $this.prev().removeClass('hide');
48             return false;
49         } else {
50             $this.prev().addClass('hide');
51             return true;
52         }
53     });
54
55     $('#password').on('blur', function() {
56         var $this = $(this);

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

59         return false;
60     } else {
61         $this.prev().addClass('hide');
62         return true;
63     }
64 });
65
66 function triggerFormVal() {
67     if ($('#userid').val().trim().length > 0 && $('#password').val().trim().length > 0) {
68         return true;
69     } else {
70         $('#userid,#password').trigger('blur');
71         return false;
72     }
73 }
74
75 $('#login').on('click', function() {
76     if (triggerFormVal()) {
77         $('#form-val').addClass('hide');
78         var resp = dash.authUser($('#userid').val().trim(), $('#password').val().trim());
79         if (resp == 1) {
80             location.href = 'dashboard.html';
81         } else if (resp == 2) {
82             location.href = 'settings.html';
83         } else if (resp == 3) {
84             alert('Invalid Userid or password');
85         } else {
86             alert('No user exists with this ID');
87         }
88     } else {
89         $('#form-val').removeClass('hide');
90     }
91 });
92 </script>
93
94 </body>
95
96 </html>

```

**Note :** We will get to `dash.authUser()` in a moment.

Save the file and run the app (`cmd + b` or `ctrl + b`). And you should see the login screen.



The page also includes simple validations.

Before we get started with the authentication logic, we will build the register screen too. Create a new file at the same level as *index.html* and name it *register.html*. Copy the below markup to *register.html*

register.html

```

1  <!DOCTYPE html>
2  <html class="bg-black">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My Dashboard | Sign up</title>
7      <meta content='width=device-width, initial-scale=1, maximum-
8      <!-- bootstrap 3.0.2 -->
9      <link href="assets/css/bootstrap.min.css" rel="stylesheet"
10     <!-- font Awesome -->
11     <link href="assets/css/font-awesome.min.css" rel="stylesheet"

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

15
16 <body class="bg-black">
17
18     <div class="form-box" id="login-box">
19         <div class="header">Register New Account</div>
20         <form>
21             <div class="body bg-gray">
22                 <div class="form-group has-error">
23                     <label class="control-label hide" id="form
24                 </div>
25                 <div class="form-group has-error">
26                     <label class="control-label hide" for="name
27                     <input type="text" name="name" id="name" c
28                 </div>
29                 <div class="form-group has-error">
30                     <label class="control-label hide" for="use
31                     <input type="text" name="userid" id="useri
32                 </div>
33                 <div class="form-group has-error">
34                     <label class="control-label hide" for="pas
35                     <input type="password" name="password" id=
36                 </div>
37             </div>
38             <div class="footer">
39                 <button type="button" id="register" class="btn
40                 <a href="index.html" class="text-center">I alr
41             </div>
42         </form>
43
44     </div>
45
46
47     <!-- jQuery 2.0.2 -->
48     <script src="assets/js/jquery.min.js"></script>
49     <script type="text/javascript">
50     $('#name').on('blur', function() {
51         var $this = $(this);
52         if ($this.val().trim().length < 1) {
53             $this.prev().removeClass('hide');
54             return false;
55         } else {
56             $this.prev().addClass('hide');
57             return true;
58         }
59     });
60
61     $('#userid').on('blur', function() {
62         var $this = $(this);
63         if ($this.val().trim().length < 1) {
64             $this.prev().removeClass('hide');
65             return false;
66         } else {
67             $this.prev().addClass('hide');
68             return true;
69         }
70     });
71     $('#password').on('blur', function() {
72         var $this = $(this);
73         if ($this.val().trim().length < 1) {

```



```

74         $this.prev().removeClass('hide');
75         return false;
76     } else {
77         $this.prev().addClass('hide');
78         return true;
79     }
80 });
81
82 function triggerFormVal() {
83     if ($('#name').val().trim().length > 0 && ($('#userid')
84         return true;
85     } else {
86         $('#name #userid #password') trigger('blur');

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

89     }
90
91     $('#register').on('click', function() {
92         if (triggerFormVal()) {
93             var resp = dash.registerUser($('#name').val().trim
94             if (resp == 1) {
95                 location.href = 'dashboard.html';
96             } else if (resp == 2) {
97                 location.href = 'settings.html';
98             } else {
99                 alert('Oops something went wrong!');
100             }
101             $('#form-val').addClass('hide');
102         } else {
103             $('#form-val').removeClass('hide');
104         }
105     });
106 </script>
107 </body>
108
109 </html>

```

**Note :** We will get to `dash.registerUser()` in a moment.

Save and run the app, you will see the login screen. At the bottom, click on the *'Register a new account'* and you should see the registration screen



with some old school jQuery validations.



Now, let us build the authentication logic. To store the user details, we will be using a node module named `diskDB`. And we will be using `bcrypt-nodejs` node package to encrypt the passwords while saving and also to compare them.



Back to terminal/prompt run

```
npm i --save diskdb bcrypt-nodejs
```

This will pull down the required dependencies.

With the power of node-webkit, you can run node modules inside your browser window. So to create an illusion that the node modules are maintained elsewhere, we will be creating a new folder named **controller** and then writing our node module related logic here. This *may be* called as **Separation of Concerns** when we are developing a node-webkit app.



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



So, create a new folder named **controller** at the root of the **development** folder. Create another folder name **collections** at the same level as **controller**. This folder will hold our JSON files. Next, create a new file named **auth.js** inside the **controller** folder. Update **auth.js** like

```
auth.js JavaScript
1 window.dash = window.dash || {};
2 var db = require('diskdb'),
3     bcrypt = require('bcrypt-nodejs');
4
5 db = db.connect('collections', ['users', 'settings']);
6
7 dash.registerUser = function(name, userid, password) {
8
9     /*
10     * response code : 0 - User with this id already exists
11     * response code : 1 - Successfully registered and User has a
12     * response code : 2 - Successfully registered and User has n
13     */
14
15     if (db.users.findOne({
16         userid: userid
17     })) return 0;
18
19     // save the user to DB
20     var savedUser = db.users.save({
21         name: name,
22         userid: userid,
23         password: bcrypt.hashSync(password)
24     });
25     return populateUser(savedUser);
26 };
27
28 dash.authUser = function(userid, password) {
29
30     /*
31     * response code : 0 - User does not exists
32     * response code : 1 - Successfully logged in and User has a
33     * response code : 2 - Successfully logged in and User has n
34     * response code : 2 - Authentication failed
35     */
36
37     // fetch the user from DB
38     var user = db.users.findOne({
39         userid: userid
40     });
41     if (user) {
42         if (bcrypt.compareSync(password, user.password)) {
43             return populateUser(user);
44         } else {
45             return 3;
46         }
47     }
48 }
```

```

47
48     } else {
49         return 0;
50     }
51 };
52
53 dash.signOut = function () {
54
55     /*
56     * Clean localStorage
57     */
58
59     localStorage.removeItem('user');
60
61 };
62
63
64 var populateUser = function(user) {
65
66     /*
67     * Create a "Session"
68     */
69
70     delete user.password; // remove password before creating a
71
72     // if local storage has a user object, the user is logged in
73     localStorage.setItem('user', JSON.stringify(user));
74
75     // check if the user has completed the settings
76     var setg = db.settings.findOne({
77         uid: user._id
78     });
79     if (setg) {
80         localStorage.setItem('settings', JSON.stringify(setg.se
81         return 1;
82     } else {
83         return 2;
84     }
85 };

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



There are quite a lot of things going on here. First, refer diskDB and bcrypt-nodejs to understand what they do.

- Line 7 : We create a register method that takes the user's name, user id and password as arguments
- Line 15 : We check if a user with same ID already exists. If yes, we return 0
- ← • Line 20 : If everything is fine, we will save the user's details in the database and call the `populateUser()` to create a "*session*".
- Line 28 : We authenticate the user by taking the credentials entered on the login page and we will respond with the appropriate codes.
- Line 64 : We save the user object in the local storage. If the object exists, it means the user is logged in.
- Line 53 : We remove the user and settings object. This will clear the *session*



Back to *register.html* add the below lines just before the end of *head* tag.

register.html

```
1 ...
2
3 <script type="text/javascript" src="controller/auth.js"></script>
4 <script type="text/javascript">
5   if (JSON.parse(localStorage.getItem('user'))) location.href =
6 </script>
```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



Save all the files and run the app. Navigate to the register screen and fill up the form and submit. If everything goes well, you should see a blank screen (*settings.html*).

Terminate the app and relaunch it and you should see the sign in screen again. Now, click on the gear icon next to the URL bar in the app (*which will be visible after you have set toolbar to true in package.json*) and you should see the dev tools come up. Click on *Resources* tab then *Local Storage* and you should see



This means that our user and session have been successfully created. Alternatively you can navigate to *development/collections/users.json* and you should see the user and the details

users.json

JavaScript

```
1 [{
2   "name": "Arvind",
3   "userid": "a",
4   "password": "$2a$10$N7dV0CxiZVfgxTf2wLS5p.TmqDPFrPkd1vV/VN3Q
5   "_id": "b3b92ebbed564996895a3a06a4d3d0f7"
6 }]
```



Sweet right?

Now, open *index.html*, we will integrate the login logic. Just before the head end tag, add the below code

```
1 ...
2
3 <script type="text/javascript" src="controller/auth.js"></script>
4 <script type="text/javascript">
5   //if (JSON.parse(localStorage.getItem('user'))) location.href =
6 </script>
7
8 </head>
9 ...
```

**Note :** We will uncomment the above line in script once the *dashboard.html* is ready.

Save all files and relaunch. Login with the registered credentials and you should be able to login. And you will see the same blank screen (*settings.html*).



Learn your way around Ionic framework. Check out Learning Ionic! user specific settings like his locations and the flick search terms. This will help us personalize the dashboard to the user. Since this is a demo app, we will capture only the said two.

[Buy](#)


At the root of *development* folder, create a new file named *settings.html*. Update it like

```
settings.html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My Dashboard | Lock screen</title>
7      <meta content='width=device-width, initial-scale=1, maximum-
8      <!-- bootstrap 3.0.2 -->
9      <link href="assets/css/bootstrap.min.css" rel="stylesheet"
10     <!-- font Awesome -->
11     <link href="assets/css/font-awesome.min.css" rel="stylesheet"
12     <!-- Theme style -->
13     <link href="assets/css/AdminLTE.css" rel="stylesheet" type="text/css"
14     <script type="text/javascript" src="controller/settings.js">
15
16 </head>
17
18 <body class="skin-blue">
19     <header class="header">
20         <a href="dashboard.html" class="logo">
21             <!-- Add the class icon to your logo image or logo
22             myDashboard
23         </a>
24         <!-- Header Navbar: style can be found in header.less
25         <nav class="navbar navbar-static-top" role="navigation"
26             <!-- Sidebar toggle button-->
27
28         </nav>
29     </header>
30     <aside class="right-side">
31         <!-- Main content -->
32         <section class="content">
33             <h3>Settings</h3>
34             <hr/>
35             <div class="col-md-6">
36                 <div class="form-group">
37                     <label for="location">My Location</label>
38                     <input type="text" class="form-control" placeholder="Location"
39                     <label class="control-label hide" for="location">
40                 </div>
41                 <div class="form-group">
42                     <label for="flickrTags">Flickr Search Tags</label>
43                     <input type="text" class="form-control" placeholder="Flickr Search Tags"
44                     <label class="control-label hide" for="flickrTags">
45                 </div>
46             </div>
47         </section>
48     </aside>
49 </body>
50 </html>
```




[Buy](#)

Learn your way around Ionic framework. Check out Learning Ionic!

```

46         <br/>
47         <button type="button" id="settings" class="btn
48         <button type="button" id="skip" class="btn bg-
49     </div>
50 </section>
51 <!-- /.content -->
52 </aside>
53
54 <!-- jQuery 2.0.2 -->
55 <script src="assets/js/jquery.min.js"></script>
56 <script type="text/javascript">
57     var settings = JSON.parse(localStorage.getItem('settings'))
58     var user = JSON.parse(localStorage.getItem('user'))
59
60     if (settings.location) $('#location').val(settings.loc
61     if (settings.flickr) $('#flickrTags').val(settings.flic
62 }
63
64
65 $('#settings').on('click', function() {
66     var $loc = $('#location').val().trim(),
67         $flickr = $('#flickrTags').val().trim();
68
69     if ($loc.length <= 0) {
70         $('#location').parents('.form-group').addClass('ha
71         $('#location').next().removeClass('hide');
72         return false;
73     } else {
74         $('#location').parents('.form-group').removeClass(
75         $('#location').next().addClass('hide');
76     }
77
78     if ($flickr.length <= 0) {
79         $('#flickrTags').parents('.form-group').addClass('h
80         $('#flickrTags').next().removeClass('hide');
81         return false;
82     } else {
83         $('#flickrTags').parents('.form-group').removeClass
84         $('#flickrTags').next().addClass('hide');
85     }
86
87     var resp = dash.saveSetting(user._id, {
88         'location': $loc,
89         'flickr': $flickr
90     });
91
92     if (resp == 1) {
93         location.href = 'dashboard.html';
94     } else {
95         alert('Oops something went wrong!');
96     }
97 });
98 $('#skip').on('click', function() {
99     if (!(settings && !settings.dummy)) {
100         localStorage.setItem('settings', JSON.stringify({
101             'dummy': 'object',
102             'location': 'new york',
103             'flickr': 'bikes'
104         }));
105     }
106     location.href = 'dashboard.html';
107 });
108 </script>
109 </body>
110
111 </html>

```



Save all files, launch the app. If you have not deleted the local storage entries, you should see the Settings page. Else login and you will see the same.

Now we will write the logic to save the user specific settings.  
Inside **controller** folder, create a new file named **settings.js**.  
Update it like

```
settings.js JavaScript
1 window.dash = window.dash || {};
2 var db = require('diskdb');
3
4 db = db.connect('collections', ['users', 'settings']);

5
6
7
8 var upd = db.settings.update({
9   uid: uid
10 }, {
11   uid: uid,
12   settings: settings
13 }, {
14   multi: false,
15   upsert: true
16 });
17
18 if (upd.updated > 0 || upd.inserted > 0) {
19   localStorage.setItem('settings', JSON.stringify(settings));
20   return 1;
21 } else {
22   return 0;
23 }
24 };
```

Learn your way around Ionic framework. Check out Learning Ionic!

Buy

Save all the files and relaunch the app. On the settings page, fill it up and click on **Save Settings**. If things go well, you should see a blank page (**dashboard.html**). Click on the back button on the toolbar and you should see the settings page with the save values and the local storage will look like



With this, we have completed the registration and login process. Now we will work on the Dashboard page.

← At the root of **development** folder, create a new file named **dashboard.html**. Update it like →

```
dashboard.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>My Dashboard | Dashboard</title>
7   <meta content='width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no' />
8   <!-- bootstrap 3.0.2 -->
9   <link href="assets/css/bootstrap.min.css" rel="stylesheet" />
10  <!-- font Awesome -->
11  <link href="assets/css/font-awesome.min.css" rel="stylesheet" />
12  <!-- Ionicons -->
13  <link href="assets/css/ionicons.min.css" rel="stylesheet" />
14  <!-- Theme style -->
15  <link href="assets/css/AdminLTE.css" rel="stylesheet" type="text/css" />
16  <style type="text/css">
17    .uname,.info,.content-header {
```

```

18     text-transform: uppercase;
19 }
20 </style>
21 <script type="text/javascript" src="controller/auth.js"></script>
22 <script type="text/javascript">
23     if (!JSON.parse(localStorage.getItem('settings'))) location
24 </script>
25 </head>
26
27 <body class="skin-blue">
28     <!-- header logo: style can be found in header.less -->
29     <header class="header">
30         <a href="dashboard.html" class="logo">

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

33     </a>
34     <!-- Header Navbar: style can be found in header.less -->
35     <nav class="navbar navbar-static-top" role="navigation">
36         <!-- Sidebar toggle button-->
37         <a href="#" class="navbar-btn sidebar-toggle" data-
38             <span class="sr-only">Toggle navigation</span>
39             <span class="icon-bar"></span>
40             <span class="icon-bar"></span>
41             <span class="icon-bar"></span>
42         </a>
43         <div class="navbar-right">
44             <ul class="nav navbar-nav">
45                 <li class="dropdown user user-menu">
46                     <a href="lockscreen.html">
47                         <i class="glyphicon glyphicon-lock">
48                         <span>LOCK SCREEN</span>
49                     </a>
50                 </li>
51                 <li class="dropdown user user-menu">
52                     <a href="settings.html">
53                         <i class="fa fa-gears"></i>
54                         <span>SETTINGS</span>
55                     </a>
56                 </li>
57                 <!-- User Account: style can be found in d
58                 <li class="dropdown user user-menu">
59                     <a href="#" class="dropdown-toggle" da
60                         <i class="glyphicon glyphicon-user">
61                         <span>
62                             <span class='uname'></span><i
63                         </span>
64                     </a>
65                     <ul class="dropdown-menu">
66                         <!-- User image -->
67                         <li class="user-header bg-light-bl
68                             
75                             <div class="pull-right">
76                                 <a href="#" id="signout" c
77                             </div>
78                         </li>
79                     </ul>
80                 </li>
81             </ul>
82         </div>
83     </nav>
84 </header>
85 <div class="wrapper row-offcanvas row-offcanvas-left">
86     <!-- Left side column. contains the logo and sidebar -
87     <aside class="left-side sidebar-offcanvas">
88         <!-- sidebar: style can be found in sidebar.less -
89         <section class="sidebar">
90             <!-- Sidebar user panel -->
91             <div class="user-panel">
92                 <div class="pull-left image">

```



```
93         
95     <div class="pull-left info">
96         <p>Hello,
97         <span class='uname'></span>
98     </p>
99     </div>
100 </div>
101 <!-- sidebar menu: : style can be found in sid
102 <ul class="sidebar-menu">
103     <li>
104         <a href="calender.html">
105             <i class="fa fa-calendar"></i>
```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```
108         </li>
109     <li>
110         <a href="settings.html">
111             <i class="fa fa-gears"></i>
112             <span>SETTINGS</span>
113         </a>
114     </li>
115 </ul>
116 </section>
117 <!-- /.sidebar -->
118 </aside>
119
120 <!-- Right side column. Contains the navbar and conten
121 <aside class="right-side">
122     <!-- Content Header (Page header) -->
123     <section class="content-header">
124         <h1>
125             Welcome to your Dashboard
126         </h1>
127     </section>
128
129     <!-- Main content -->
130     <section class="content">
131         <div class="box box-info">
132             <div class="box-header">
133                 <h3 class="box-title">MY WEATHER REPORT</h3>
134                 <div class="box-tools pull-right">
135                     <div class="label bg-aqua maploc">
136                         </div>
137                 </div>
138                 <div class="box-body">
139                     <div id="weather"></div>
140                     <!-- /.box-body -->
141                 </div>
142                 <div class="box-footer">
143                     <a href="settings.html">Change location</a>
144                 </div>
145             </div>
146             <div class="box box-warning">
147                 <div class="box-header">
148                     <h3 class="box-title">MY MAP</h3>
149                     <div class="box-tools pull-right">
150                         <div class="label bg-aqua maploc">
151                             </div>
152                     </div>
153                     <div class="box-body">
154                         <div id="map-canvas"></div>
155                         <!-- /.box-body -->
156                     </div>
157                     <div class="box-footer">
158                         <a href="settings.html">Change location</a>
159                     </div>
160                 </div>
161             <div class="box box-danger">
162                 <div class="box-header">
163                     <h3 class="box-title">MY FLICKR SEARCH</h3>
164                     <div class="box-tools pull-right">
165                         <div class="label bg-aqua flickrta<
166                             </div>
167                     </div>
```



```

168         <div class="box-body">
169             <div id="wrapper">
170                 <div id="columns">
171                     </div>
172                 </div>
173                 <!-- /.box-body -->
174                 <div class="box-footer">
175                     <a href="settings.html">Change tag
176                 </div>
177             </div>
178         </div>
179     </section>
180 <!-- / content -->

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

183     </div>
184     <!-- ./wrapper -->
185
186     <!-- jQuery 2.0.2 -->
187     <script src="assets/js/jquery.min.js"></script>
188     <!-- Bootstrap -->
189     <script src="assets/js/bootstrap.min.js" type="text/javasc
190     <script src="https://maps.googleapis.com/maps/api/js?v=3.e
191     <!-- AdminLTE App -->
192     <script src="assets/js/app.js" type="text/javascript"></sc
193     <script type="text/javascript">
194     var user = JSON.parse(localStorage.getItem('user'));
195     var settings = JSON.parse(localStorage.getItem('settings')
196     $('#uname').text(user.name.trim());
197
198     $('#signout').on('click', function(e) {
199         e.preventDefault();
200         if (dash.signOut()) location.href = 'index.html';
201         else alert('Oops something went wrong!');
202     });
203
204     // load widgets async
205     $.get('widgets/flickr.html', function(data) {
206         $('#body').append(data)
207     });
208     $.get('widgets/maps.html', function(data) {
209         $('#body').append(data);
210     });
211     $.get('widgets/weather.html', function(data) {
212         $('#body').append(data);
213     });
214
215     // update the dashboard with values from settings
216     $('#maploc').text(settings.location.trim().toUpperCase());
217     $('#flickrtags').text(settings.flickr.trim().toUpperCase()
218     </script>
219 </body>
220
221 </html>

```



Standard markup and scripts. You will understand as you read the file. Line nos 205, 208 and 211 refer our widgets, which we will create soon. Line 198 registers the Sign Out and calls the method in *auth.js*.

When you save all the files and run, you should see



Awesome! Our dashboard structure is ready.

Next, we will implement the lock screen. If the user wants to lock the app, he can do so with this feature. The lock screen does not really make sense here, but you can see the kind of options this feature opens up.

Create a new file named *lockscreen.html* inside the *development* folder and fill it up like



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

1  <!DOCTYPE html>
2  <html class="lockscreen">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My Dashboard | Lock screen</title>
7      <meta content='width=device-width, initial-scale=1, maximum-
8      <!-- bootstrap 3.0.2 -->
9      <link href="assets/css/bootstrap.min.css" rel="stylesheet"
10     <!-- font Awesome -->
11     <link href="assets/css/font-awesome.min.css" rel="stylesheet"
12     <!-- Theme style -->
13     <link href="assets/css/AdminLTE.css" rel="stylesheet" type="text/css"
14     <script type="text/javascript">
15     if (!JSON.parse(localStorage.getItem('user'))) location.href = "#/login";
16     </script>
17     <script type="text/javascript" src="controller/auth.js"></script>
18
19 </head>
20
21 <body>
22     <!-- Automatic element centering using js -->
23     <div class="">
24         <div class="headline text-center" id="time">
25             <!-- Time auto generated by js -->
26         </div>
27         <!-- /.headline -->
28
29         <!-- User name -->
30         <div class="lockscreen-name"></div>
31
32         <!-- START LOCK SCREEN ITEM -->
33         <div class="lockscreen-item">
34             <!-- lockscreen image -->
35             <div class="lockscreen-image">
36                 
37             </div>
38             <!-- /.lockscreen-image -->
39
40             <!-- lockscreen credentials (contains the form) -->
41             <div class="lockscreen-credentials">
42
43                 <div class="input-group">
44                     <input type="password" class="form-control" />
45                     <div class="input-group-btn">
46                         <button class="btn btn-flat" id="relogin">Log in</button>
47                     </div>
48                 </div>
49             </div>
50             <!-- /.lockscreen credentials -->
51
52         </div>
53         <!-- /.lockscreen-item -->
54
55         <div class="lockscreen-link">
56             <a href="javascript:" id="logout">Or sign in as a guest</a>
57         </div>
58     </div>
59 </body>

```


[Buy](#)

Learn your way around Ionic framework. Check out Learning Ionic!

```

60     <!-- /.center -->
61
62     <!-- jQuery 2.0.2 -->
63     <script src="assets/js/jquery.min.js"></script>
64
65     <!-- page script -->
66     <script type="text/javascript">
67     var user = JSON.parse(localStorage.getItem('user'));
68     $(function() {
69         startTime();
70         $('#lockscreen-name').text(user.name.trim());
71     });
72
73
74
75         if (dash.signOut()) location.href = 'index.html';
76         else alert('Oops something went wrong!');
77     })
78
79     $('#relogin').on('click', function() {
80         var $pwd = $('#password').val().trim();
81         if ($pwd.length == 0) {
82             alert('Please enter a valid password');
83         } else {
84             var resp = dash.authUser(user.userid, $pwd);
85             if (resp == 1) {
86                 location.href = 'dashboard.html';
87             } else if (resp == 2) {
88                 location.href = 'settings.html';
89             } else if (resp == 3) {
90                 alert('Invalid password');
91             } else {
92                 alert('Oops something went wrong');
93             }
94         }
95     });
96
97     /* */
98     function startTime() {
99         var today = new Date();
100         var h = today.getHours();
101         var m = today.getMinutes();
102         var s = today.getSeconds();
103
104         // add a zero in front of numbers<10
105         m = checkTime(m);
106         s = checkTime(s);
107
108         //Check for PM and AM
109         var day_or_night = (h > 11) ? "PM" : "AM";
110
111         //Convert to 12 hours system
112         if (h > 12)
113             h -= 12;
114
115         //Add time to the headline and update every 500 millis
116         $('#time').html(h + ":" + m + ":" + s + " " + day_or_n);
117         setTimeout(function() {
118             startTime()
119         }, 500);
120     }
121
122     function checkTime(i) {
123         if (i < 10) {
124             i = "0" + i;
125         }
126         return i;
127     }
128 </script>
129 </body>
130
131 </html>

```





Next, open *index.html* and uncomment the line in script tag we have commented earlier.

Save all the files, launch the app, click on Lock Screen on the dashboard page and you should see



Now, enter the password and click on the screen. If the

Learn your way around Ionic framework. Check out Learning Ionic!

Buy



click on "Sign in as a different user" and the current user will be logged off and you can login again with a different user.

Now, you can start feeling the potential of node webkit. The things which you can do with it! *Hey! I was excited too..*

Next, we will create the widgets on the page. Create a new folder named *widgets* inside the *development* folder. We will place each widget here to keep things a bit structured.

Create a new file named *weather.html*. This widget will contact the Yahoo weather service and get the data for the location configured in the settings. If the user has not set any location, we will use 'New York'.

This widget show cases how you can integrate a node module with a UI. Open *weather.html* and add the below code

```
weather.html
1 <style type="text/css">
2 .wh-wrapper {
3     background-color: #f0f7fd;
4     border: 1px solid #d0e3f0;
5     padding: 10px;
6     text-transform: uppercase;
7 }
8 .wh-wrapper .col-md-3 {
9     text-align: center;
10    border-right: 1px solid #64A4CF;
11 }
12 .wh-wrapper .col-md-3:last-child {
13     border-right: none;
14 }
15 .wh-wrapper .col-md-3 h4 {
16     padding-bottom: 5px;
17     border-bottom: 1px solid #00c0ef;
18 }
19 .wh-wrapper h2{
20     text-align: center;
21 }
22 </style>
23 <script type="text/javascript" src="controller/weather.js"></script>
24 <script type="text/javascript">
```

```

25 var settings = JSON.parse(localStorage.getItem('settings'));
26 dash.getWeather(settings.location, function(data) {
27     console.log(data);
28     if (data.length > 0) {
29         var wthr = data[0].current; // show the current data fr
30         var html = '<div class="wh-wrapper">';
31         html += '<h3>';
33         html += '<div class="col-md-3">';
34         html += '<h4>Today</h4><label>'+wthr.da
35         html += '</div>';
36         html += '<div class="col-md-3">';
37         html += '<h4>Temperature</h4><label>'+wthr

```

Learn your way around Ionic framework. Check out Learning Ionic!

Buy

X

```

40         html += '<h4>Wind</h4><label>'+wthr.win
41         html += '</div>';
42         html += '<div class="col-md-3">';
43         html += '<h4>Humidity</h4><label>'+wthr
44         html += '</div>';
45         html += '</div>';
46         html += '</div>';
47
48         $('#weather').html(html);
49     } else {
50         // no data
51         $('#weather').html('<div class="wh-wrapper"><h2>Service
52     }
53 });
54 </script>

```

Some styles and markup. The data will be coming from a node module named *weather-js*. Back to terminal/prompt run

```
npm i --save weather-js
```

Next, create a new file named *weather.js* inside the *controller* folder and add the below code.

```

weather.js JavaScript
1 window.dash = window.dash || {};
2 var weather = require('weather-js');
3
4 dash.getWeather = function(location, callback) {
5     weather.find({
6         search: location
7     }, function(err, result) {
8         if (err) console.log(err);
9
10        callback(result);
11    });
12 }

```

When the widget is injected into the dashboard, the script inside the *weather.html* will call the `getWeather()` and the response will be parsed and a HTML will be built.

Save all the files and run. I have set my location to Hyderabad and this is how my dashboard looks



Sweet! and it is drizzling outside!

Now, create a new file named *maps.html* inside the *widgets* folder. This will consist of the logic to display the map.  
Update the file like



maps.html

```
1 <style type="text/css">
2
3
4     margin: 0px;
5     padding: 0px
6 }
7 </style>
8 <script type="text/javascript">
9 var settings = JSON.parse(localStorage.getItem('settings')),
10     geocoder, map;
11
12 function initialize() {
13     var address = settings.location;
14     geocoder = new google.maps.Geocoder();
15     geocoder.geocode({
16         'address': address
17     }, function(results, status) {
18         if (status == google.maps.GeocoderStatus.OK) {
19             var mapOptions = {
20                 zoom: 8,
21                 center: results[0].geometry.location
22             }
23             map = new google.maps.Map(document.getElementById('map'), mapOptions);
24             var marker = new google.maps.Marker({
25                 map: map,
26                 position: results[0].geometry.location
27             });
28         } else {
29             alert('Geocode was not successful for the following address: ' + address);
30         }
31     });
32 }
33 google.maps.event.addDomListener(window, 'load', initialize);
34 </script>
```

Learn your way around Ionic framework. Check out Learning Ionic!

Buy



In this file, we will fetch the location from the localstorage, pass it to geocoder service and get the position. Then we will generated a map based on this position.



Save all the files and run the app. You should see



Finally, we will get a few images from flickr and display it on the dashboard to make it complete. Create a new file named *flickr.html* inside the *widgets* folder and update it like

flickr.html

```
1 <style type="text/css">
2 #wrapper {
3     width: 90%;
4     max-width: 700px;
5     min-width: 500px;
6     margin: 50px auto;
7     margin-top: 0px;
```

```

8     margin-bottom: 0px;
9 }
10 #columns {
11     -webkit-column-count: 3;
12     -webkit-column-gap: 10px;
13     -webkit-column-fill: auto;
14     -moz-column-count: 3;
15     -moz-column-gap: 10px;
16     -moz-column-fill: auto;
17     column-count: 3;
18     column-gap: 15px;
19     column-fill: auto;
20 }

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

23     background: #FEFEFE;
24     border: 2px solid #FAFAFA;
25     box-shadow: 0 1px 2px rgba(34, 25, 25, 0.4);
26     margin: 0 2px 15px;
27     -webkit-column-break-inside: avoid;
28     -moz-column-break-inside: avoid;
29     column-break-inside: avoid;
30     padding: 15px;
31     padding-bottom: 5px;
32     background: -webkit-linear-gradient(45deg, #FFF, #F9F9F9);
33     opacity: 1;
34     -webkit-transition: all .2s ease;
35     -moz-transition: all .2s ease;
36     -o-transition: all .2s ease;
37     transition: all .2s ease;
38 }
39 .pin img {
40     width: 100%;
41     padding-bottom: 15px;
42     margin-bottom: 5px;
43 }
44 </style>
45
46 <script>
47 var settings = JSON.parse(localStorage.getItem('settings'));
48 $.getJSON("http://api.flickr.com/services/feeds/photos_public.g
49     tags: settings.flickr,
50     tagmode: "any",
51     format: "json"
52 },
53 function(data) {
54     var imgs = '';
55     $.each(data.items, function(i, item) {
56         imgs += '<div class="pin">
6      <title>My Dashboard | Dashboard</title>
7      <meta content='width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no' />
8      <!-- bootstrap 3.0.2 -->
9      <link href="assets/css/bootstrap.min.css" rel="stylesheet" />
10     <!-- font Awesome -->
11     <link href="assets/css/font-awesome.min.css" rel="stylesheet" />

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

15     <link href="assets/css/fullcalendar.css" rel="stylesheet" />
16     <!-- Theme style -->
17     <link href="assets/css/AdminLTE.css" rel="stylesheet" type="text/css" />
18     <style type="text/css">
19         .uname, .info, .content-header {
20             text-transform: uppercase;
21         }
22     </style>
23     <script type="text/javascript" src="controller/auth.js"></script>
24     <script type="text/javascript">
25         if (!JSON.parse(localStorage.getItem('settings'))) location.href = 'login.html';
26     </script>
27 </head>
28
29 <body class="skin-blue">
30     <!-- header logo: style can be found in header.less -->
31     <header class="header">
32         <a href="dashboard.html" class="logo">
33             <!-- Add the class icon to your logo image or logo font to replace the class 'logo' -->
34             myDashboard
35         </a>
36         <!-- Header Navbar: style can be found in header.less -->
37         <nav class="navbar navbar-static-top" role="navigation">
38             <!-- Sidebar toggle button -->
39             <a href="#" class="navbar-btn sidebar-toggle" data-toggle="offcanvas" data-target="#sidebar" data-
40                 <span class="sr-only">Toggle navigation</span>
41                 <span class="icon-bar"></span>
42                 <span class="icon-bar"></span>
43                 <span class="icon-bar"></span>
44             </a>
45             <div class="navbar-right">
46                 <ul class="nav navbar-nav">
47                     <li class="dropdown user user-menu">
48                         <a href="lockscreen.html">
49                             <i class="glyphicon glyphicon-lock"></i>
50                             <span>LOCK SCREEN</span>
51                         </a>
52                     </li>
53                     <li class="dropdown user user-menu">
54                         <a href="settings.html">
55                             <i class="fa fa-gears"></i>
56                             <span>SETTINGS</span>
57                         </a>
58                     </li>
59                     <!-- User Account: style can be found in dropdown-user.less -->
60                     <li class="dropdown user user-menu">
61                         <a href="#" class="dropdown-toggle" data-toggle="dropdown">
62                             <i class="glyphicon glyphicon-user"></i>
63                             <span></span>
64                             <span class='uname'></span><i class="fa fa-user"></i>
65                         </a>
66                         <ul class="dropdown-menu">
67                             <!-- User image -->
68                             <li class="user-header bg-light-blue">
69                                 
70                                 <p>
71                                     <span class='uname'></span><i class="fa fa-user"></i>
72                                 </p>
73

```



```

74         </li>
75         <!-- Menu Footer-->
76         <li class="user-footer">
77             <div class="pull-right">
78                 <a href="#" id="signout" c
79             </div>
80         </li>
81     </ul>
82 </li>
83 </ul>
84 </div>
85 </nav>
86 </header>

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



```

89     <aside class="left-side sidebar-offcanvas">
90         <!-- sidebar: style can be found in sidebar.less -
91         <section class="sidebar">
92             <!-- Sidebar user panel -->
93             <div class="user-panel">
94                 <div class="pull-left image">
95                     
97                 <div class="pull-left info">
98                     <p>Hello,
99                         <span class='uname'></span>
100                     </p>
101                 </div>
102             </div>
103             <!-- sidebar menu: : style can be found in sid
104             <ul class="sidebar-menu">
105                 <li>
106                     <a href="calender.html">
107                         <i class="fa fa-calendar"></i>
108                         <span>CALENDER</span>
109                     </a>
110                 </li>
111                 <li>
112                     <a href="settings.html">
113                         <i class="fa fa-gears"></i>
114                         <span>SETTINGS</span>
115                     </a>
116                 </li>
117             </ul>
118         </section>
119         <!-- /.sidebar -->
120     </aside>
121
122     <!-- Right side column. Contains the navbar and conten
123     <aside class="right-side">
124         <!-- Content Header (Page header) -->
125         <section class="content-header">
126             <h1>
127                 Calendar
128                 <small>Your appointments</small>
129             </h1>
130             <ol class="breadcrumb">
131                 <li><a href="index.html"><i class="fa fa-h
132                 </li>
133                 <li class="active">Calendar</li>
134             </ol>
135         </section>
136
137         <!-- Main content -->
138         <section class="content">
139
140
141             <div class="row">
142                 <div class="col-md-3">
143                     <div class="box box-primary">
144                         <div class="box-header">
145                             <h4 class="box-title">Draggabl
146                         </div>
147                         <div class="box-body">
148                             <!-- the events -->

```





Learn your way around Ionic framework. Check out Learning Ionic!

Buy

```
149         <div id='external-events'>
150             <div class='external-event
151             <div class='external-event
152             <div class='external-event
153             <div class='external-event
154             <div class='external-event
155             <p>
156                 <input type='checkbox'
157                 <label for='drop-remov
158             </p>
159         </div>
160     </div>
161     <!-- /.box-body -->
```

```
164         <div class="box box-primary">
165             <div class="box-header">
166                 <h3 class="box-title">Create E
167             </div>
168             <div class="box-body">
169                 <div class="btn-group" style="
170                     <button type="button" id="
171                         <span class="caret"></
172                     </button>
173                     <ul class="dropdown-menu"
174                         <li><a class="text-gre
175                         </li>
176                         <li><a class="text-bl
177                         </li>
178                         <li><a class="text-nav
179                         </li>
180                         <li><a class="text-yel
181                         </li>
182                         <li><a class="text-ora
183                         </li>
184                         <li><a class="text-aqu
185                         </li>
186                         <li><a class="text-red
187                         </li>
188                         <li><a class="text-fuc
189                         </li>
190                         <li><a class="text-pur
191                         </li>
192                     </ul>
193                 </div>
194                 <!-- /.btn-group -->
195                 <div class="input-group">
196                     <input id="new-event" type=
197                     <div class="input-group-bt
198                         <button id="add-new-ev
199                     </div>
200                 <!-- /.btn-group -->
201             </div>
202         <!-- /.input-group -->
203     </div>
204 </div>
205 </div>
206 <!-- /.col -->
207 <div class="col-md-9">
208     <div class="box box-primary">
209         <div class="box-body no-padding">
210             <!-- THE CALENDAR -->
211             <div id="calendar"></div>
212         </div>
213         <!-- /.box-body -->
214     </div>
215     <!-- /. box -->
216 </div>
217 <!-- /.col -->
218 </div>
219 <!-- /.row -->
220 </section>
221 <!-- /.content -->
222 </aside>
223 <!-- /.right-side -->
```




[Buy](#)

Learn your way around Ionic framework. Check out Learning Ionic!

```

224 </div>
225 <!-- ./wrapper -->
226
227 <!-- jQuery 2.0.2 -->
228 <script src="assets/js/jquery.min.js"></script>
229 <!-- jQuery UI 1.10.3 -->
230 <script src="assets/js/jquery-ui-1.10.3.min.js" type="text
231 <!-- Bootstrap -->
232 <script src="assets/js/bootstrap.min.js" type="text/javasc
233 <!-- AdminLTE App -->
234 <script src="assets/js/app.js" type="text/javascript"></sc
235 <!-- fullCalendar -->
236 <script src="assets/js/fullcalendar.min.js" type="text/jav

```

```

239 $('username').text(user.name.trim());
240
241 $('#signout').on('click', function(e) {
242     e.preventDefault();
243     if (dash.signOut()) location.href = 'index.html';
244     else alert('Oops something went wrong!');
245 });
246
247 $(function() {
248     /* initialize the external events
249     -----
250     function ini_events(ele) {
251         ele.each(function() {
252
253             // create an Event Object (http://arshaw.com/fr
254             // it doesn't need to have a start or end
255             var eventObject = {
256                 title: $.trim($(this).text()) // use the e
257             };
258
259             // store the Event Object in the DOM element s
260             $(this).data('eventObject', eventObject);
261
262             // make the event draggable using jQuery UI
263             $(this).draggable({
264                 zIndex: 1070,
265                 revert: true, // will cause the event to g
266                 revertDuration: 0 // original position af
267             });
268         });
269     });
270
271     }
272     ini_events($('#external-events div.external-event'));
273
274     /* initialize the calendar
275     -----
276     //Date for the calendar events (dummy data)
277     var date = new Date();
278     var d = date.getDate(),
279         m = date.getMonth(),
280         y = date.getFullYear();
281     $('#calendar').fullCalendar({
282         header: {
283             left: 'prev,next today',
284             center: 'title',
285             right: 'month,agendaWeek,agendaDay'
286         },
287         buttonText: { //This is to add icons to the visibl
288             prev: "<span class='fa fa-caret-left'></span>",
289             next: "<span class='fa fa-caret-right'></span>",
290             today: 'today',
291             month: 'month',
292             week: 'week',
293             day: 'day'
294         },
295         //Random default events
296         events: [{
297             title: 'All Day Event',
298             start: new Date(y, m, 1),

```



Learn your way around Ionic framework. Check out Learning Ionic!

Buy

```

299         backgroundColor: "#f56954", //red
300         borderColor: "#f56954" //red
301     }, {
302         title: 'Long Event',
303         start: new Date(y, m, d - 5),
304         end: new Date(y, m, d - 2),
305         backgroundColor: "#f39c12", //yellow
306         borderColor: "#f39c12" //yellow
307     }, {
308         title: 'Meeting',
309         start: new Date(y, m, d, 10, 30),
310         allDay: false,
311         backgroundColor: "#0073b7" //Blue
312     }, {
313         title: 'Lunch',
314         start: new Date(y, m, d, 12, 0),
315         end: new Date(y, m, d, 14, 0),
316         allDay: false,
317         backgroundColor: "#00c0ef", //Info (aqua)
318         borderColor: "#00c0ef" //Info (aqua)
319     }, {
320         title: 'Birthday Party',
321         start: new Date(y, m, d + 1, 19, 0),
322         end: new Date(y, m, d + 1, 22, 30),
323         allDay: false,
324         backgroundColor: "#00a65a", //Success (green)
325         borderColor: "#00a65a" //Success (green)
326     }
327 ],
328 editable: true,
329 droppable: true, // this allows things to be dropped
330 drop: function(date, allDay) { // this function is
331
332     // retrieve the dropped element's stored Event
333     var originalEventObject = $(this).data('eventObject');
334
335     // we need to copy it, so that multiple events
336     var copiedEventObject = $.extend({}, originalEventObject);
337
338     // assign it the date that was reported
339     copiedEventObject.start = date;
340     copiedEventObject.allDay = allDay;
341     copiedEventObject.backgroundColor = $(this).css("background-color");
342     copiedEventObject.borderColor = $(this).css("border-color");
343
344     // render the event on the calendar
345     // the last `true` argument determines if the event is
346     $( '#calendar' ).fullCalendar('renderEvent', copiedEventObject, true);
347
348     // is the "remove after drop" checkbox checked?
349     if ( $( '#drop-remove' ).is(':checked') ) {
350         // if so, remove the element from the "Draggable"
351         $(this).remove();
352     }
353
354     }
355 });
356
357 /* ADDING EVENTS */
358 var currColor = "#f56954"; //Red by default
359 //Color chooser button
360 var colorChooser = $( "#color-chooser-btn" );
361 $( "#color-chooser > li > a" ).click(function(e) {
362     e.preventDefault();
363     //Save color
364     currColor = $(this).css("color");
365     //Add color effect to button
366     colorChooser
367         .css({
368             "background-color": currColor,
369             "border-color": currColor
370         })
371         .html($(this).text() + ' <span class="caret"><
372 });
373 $( "#add-new-event" ).click(function(e) {

```

```

374     e.preventDefault();
375     //Get value and make sure it is not null
376     var val = $("#new-event").val();
377     if (val.length == 0) {
378         return;
379     }
380
381     //Create event
382     var event = $("

Learn your way around Ionic framework. Check out Learning Ionic!



Buy



```

389         $('#external-events').prepend(event);
390
391         //Add draggable functionality
392         ini_events(event);
393
394         //Remove event from text input
395         $("#new-event").val("");
396     });
397 });
398 </script>
399 </body>
400
401 </html>

```



Save all the files and run the app. You should see



This completes our Desktop dashboard. Simple sweet and awesome!



## Distributing the application



Our app is ready to be distributed. You can follow the steps here to build an installer. Do not forget to set "toolbar": false and clean up the data before publishing.



Hope you got an idea as how to build node webkit apps easily.



Thanks for reading! Do comment.  
@arvindr21



Series Navigation



<< Node Webkit – Build Desktop Apps with Node and Web Technologies



Node webkit and Angularjs – A MovieStub App >>



http://thejackalofjavascript.com/node-webkit-powered-dashboard-app/



32/37


```

Like this:

★ Like

Be the first to like this.

## Related

Onsen UI and

Building a Todo App

Deploying Node

Learn your way around Ionic framework. Check out Learning Ionic!

Buy

May 9, 2014  
In "Android"

In "Angularjs"

May 17, 2014  
In "Deployments"

 JULY 20, 2014


 ARVIND RAVULAVARU

25 Comments

The Jackal of Javascript

 Login

 Recommend 3

 Share

Sort by Best



Join the discussion...



**Angelo V. Merlo** • 6 months ago

first thank you for the great tutorial, I would like to help me, as I do to start the application with the window maximized? Thank you very much!

^ | v • Reply • Share >



**drzed** • 7 months ago

Hey Arvind, thanks a million for doing these great tutorials. Just curious, have you tried (or is it possible) to run a Node WebKit app in docker - using containerisation?

^ | v • Reply • Share >



**Nelson Rodriguez** • 9 months ago

Hi and thank a lot for sharing! Thinking in phonegap app what will be my procedure to achieve the même project? When you say wrapping th main template in cordova what do you mens exactly? I saw you other tutorial using framework7 to phonegap but it's not the same case.

And what about using vue.js? It will compile in phonegap case?

Thanks a lot!

^ | v • Reply • Share >

**Pradeep Snnakar** • 9 months ago

I have created a nwjs application and I have created the distribution package using nw-builder as `sudo nwbuild -p linux -v 0.12.3 -o ~/Documents/output/ /home/path/to/application/files/` and it was created successfully. I used diskdb for data storage by following ur tutorial series.

The problem is, when i save some data records and close the application and next time when i open the application, previously saved data is unavailable..! When I made toolbar: true in my package.json file i observed that each time the chromium webkit open as a temporary folder with temp

Learn your way around Ionic framework. Check out Learning Ionic!

[Buy](#)

file:///tmp/.org.chromium.Chromium.dBISrk/login.nimi etc. How can I solve the same?

^ | v • Reply • Share ›

**Brian Schermerhorn** • 10 months ago

How would one interact with the native features of an OS? This mostly just feels like it's a web app on a desktop. Are there limitations to what you can do with NW?

Thanks for the tutorial =)

^ | v • Reply • Share ›

**Jonathan Willian** • a year ago

Well, at first, THANK YOU SO MUCH!

This really save my life, for real!

I'm starting coding with NW.js and your article helped a lot, it's a very well done tutorial.

Now I can do a lot of thing just because of you, so thanks again!

Keep the good work, sir, and sorry about the bad English.

^ | v • Reply • Share ›

**Arvind Ravulavaru** Mod → Jonathan Willian • a year ago

Thanks Jonathan!

^ | v • Reply • Share ›

**He Zhang** • a year ago

Thanks for sharing. It looks awesome! One thing I am confused is that I see you use diskdb module to store user data. But it's local storage in each end-user's PC, right? So does this mean if someone sign up in one PC but he can't login in another PC? In real product should we use something like MongoDB hosted in cloud to store user data? Sorry I'm pretty new to desktop apps.

Thank you in advance!

^ | v • Reply • Share ›

**Arvind Ravulavaru** Mod → He Zhang • a year ago

Hello He Zhang. Yes, you need to use MongoDB for those purposes. But do remember that this may be used offline also, so you may not have access to MongoDB all the time. So for that you can use the online/offline event to detect if the user is online/offline and if the user is offline, you can save data on the user's machine/diskDB. And once he goes back online you can sync it

with MongoDB.

The main challenge here is password management. If the user is offline, you should be still able to log the user in. So, you need to decide, if diskDB should store a copy of the user's credentials. This is a design decision you need to make.

Thanks.

^ | v • Reply • Share ›



Learn your way around Ionic framework. Check out Learning Ionic!

Buy



Great tutorial, and thank you for it. But will be glad to see a bit of the implementation of the online/offline event.

^ | v • Reply • Share ›



**Viktor Trush** • a year ago



Awesome tutorial.

Would it be possible to distribute this application to Adroid and iOS devices?

^ | v • Reply • Share ›



**Arvind Ravulavaru** Mod → Viktor Trush • a year ago



Thanks Viktor. Not directly. You can wrap the main template in Cordova to achieve that.

^ | v • Reply • Share ›



**Abdullah Almsaeed** • 2 years ago



Amazing use of AdminLTE! Thanks for sharing such an amazing tutorial.

^ | v • Reply • Share ›



**Arvind Ravulavaru** Mod → Abdullah Almsaeed • 2 years ago



Thanks Abdullah! Thank you for open sourcing this awesome template.

^ | v • Reply • Share ›



**Jagdeep Sahdeva** • 2 years ago



Hi Arvind Thanks for sharing. I noticed that the index.html is missing `<script type="text/javascript" src="controller/auth.js"></script>`. This should be just after the `</head>` tag. This will fix the error reported by Monique.

^ | v • Reply • Share ›



**Arvind Ravulavaru** Mod → Jagdeep Sahdeva • 2 years ago



Thanks **@Jagdeep Sahdeva** !

The final version of the code is having the include inside the head tag : <https://github.com/arvindr21/m...>

Looking at Monique's project structure, I figured this was the issues.

Thanks.

^ | v • Reply • Share ›

**Monique** • 2 years ago

Hello Arvind, I have run into a problem following the tutorial, I am getting an error after adding in the auth.js file. Before that point the app ran perfectly. So after adding the authorisation logic I relaunched the app, clicked on register and when I try and create an account I get this error:

TypeError: Cannot read property 'findOne' of undefined  
at Object.dash.registerUser

Learn your way around Ionic framework. Check out Learning Ionic!

[Buy](#)

:) I am still learning

^ | v • Reply • Share ›

**Arvind Ravulavaru** Mod → Monique • 2 years ago

Hello Monique, this error is because you have not installed/included diskdb node module. You need to run

```
$ npm i diskdb
```

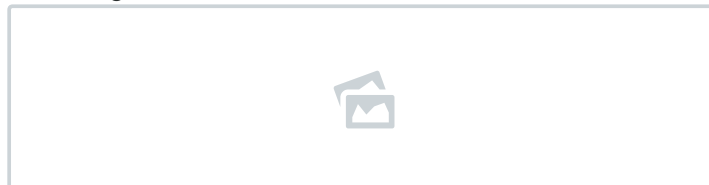
This should solve the issue.

Thanks.

^ | v • Reply • Share ›

**Monique** → Arvind Ravulavaru • 2 years ago

This is my folder structure (attached). I ran the npm i diskdb again but still errors out



^ | v • Reply • Share ›

**Arvind Ravulavaru** Mod → Monique • 2 years ago

Hello Monique, HTML and JSON files need to be outside the node\_modules folder at the root of the development folder.. Please refer :

<https://github.com/arvindr21/m...>

^ | v • Reply • Share ›

**Shoalal** → Arvind Ravulavaru • 2 years ago

Hi Arvind,

First i'd like to say thank you so much for sharing this with us, much appreciate!

I am experiencing the same error as Monique and I would really like your assistance in solving the "Cannot read property 'findOne' of undefined" problem.

As per your suggestion, I have all my files (HTML and JSON) outside the node\_modules folder.



However am still getting the same error.

Any suggestion?

Kind Regards,

Shoalai

^ | v • Reply • Share ›




 **Arvind Ravulavaru** Mod → Shoalai • 2 years ago — | 📌

Learn your way around Ionic framework. Check out Learning Ionic!  
your folder structure, I can help you better.

[Buy](#)



^ | v • Reply • Share ›

 **Gerald** → Arvind Ravulavaru • 7 months ago — | 📌  
Hi Arvind, I'm getting the same error as Shoalai. Plz enlighten me on how to interact with the diskDB. It works fine when launching from sublime but not from installer created by nodebob.

