

The Jackal of Javascript

Knowing is different & Doing is different

[HOME](#)[ABOUT ME](#)[HISTORY OF THE WEB](#)[JAVASCRIPT 101](#)[REQUEST](#)

Node Webkit – Build Desktop Apps with Node and Web Technologies

This post is part 1 of 7 in the series [Node Webkit Apps](#)



Looking for a real time implementation? Check
: [Node Webkit powered Dashboard application](#)

In this post, we will see how to build cross OS desktop

Learn your way around Ionic framework. Check out [Learning Ionic!](#) a look at building and running a simple node webkit app and then expand on it by adding different features like platform services, node modules etc.

You can find the code for the examples in this post [here](#).

So, let us get started.

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

Buy



Linked In

Google Plus

Mongo Director

Site Point

Contents

What is node webkit
 Setting up node webkit
 Hello node webkit
 Run the application
 Some more config options
 Adding Native features
 Using Node Modules
 Distribute the App
 App Security

Chronicles of the
 Jackal of Javascript

Chrome App

Chrome Extension

Workshops

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

← What is node webkit

node-webkit is an app runtime based on Chromium and node.js. You can write native apps in HTML and JavaScript with node-webkit. It also lets you call Node.js modules directly from the DOM and enables a new way of writing native applications with all Web technologies.

It's created and developed in the Intel Open Source Technology Center.

Presentations

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



Setting up node webkit

To build a desktop app using node webkit, we will need to get the prebuilt binaries. Then we will pass our application to the node webkit lib to run the app.

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

Buy



depending on your OS. Now, create a new folder named **myNodeWebkitApps** and unzip the download here. To keep things simple, rename the unzipped folder to **nw**.

Hello node webkit

- Rotten Tomatoes jQuery Rest Client
- blueimp-file-upload-expressjs - A Node Package
- lazyboy - Command line laziness redefined

Now, we will build a “hello world” app. Create another folder named *helloWorld* inside *myNodeWebKitApps* folder. The folder structure so far would be



Next, open a terminal/prompt inside the *helloWorld* folder and run

```
npm init
```

and fill it up like



This will generate a standard *package.json* file. To run a node webkit app, you need only 2 properties

1. main – The first file that needs to be executed when the app is launched
2. name -The name of the package. This **must** be a unique, lowercase alpha-numeric name without spaces. It may include “.” or “_” or “-” characters. It is otherwise opaque.

As we have mentioned in the *package.json*, our starter file would be *index.html*. Create a new file named *index.html* at the root of *helloWorld* folder and open it up in your favorite editor. We will be displaying “Hello Node Webkit” in a header tag as the final output. Add the below code to *index.html*

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



```
1 <html>
2 <head>
3   <title>Hello Node Webkit</title>
4 </head>
5 <body>
6   <h1>Hello Node Webkit</h1>
7 </body>
8 </html>
```

That is all we need to run our app.

Run the application

- [diskDB](#) - A Light Weight Disk based JSON Database - A Node Package
- [My Slush Generators](#)
- [My Yeoman Generators](#)
- [My Web Components](#)
- [My Bower Packages](#)

Series

- [Ionic, Twilio and Node Scheduler – A Reminder App](#)
- [Node Webkit Apps](#)
- [Raspberry Pi](#)

Guest Posts

- [Building a App with webkit, Firebase, !\[\]\(94c04a5e8e7b3269dd6cea1792ca5aaf_img.jpg\)](#)
- [Creating a Sentiment Analysis Application Using Node.js](#)
- [Implementing pagination with MongoDB, Express.js & Slush](#)

To run the application, we need to zip (*not rar or 7z*) files (*not the folder itself*) and rename the file to *app.nw*.

Note : The *nw* app name *SHOULD* be in lowercase.

Back to the *helloWorld* folder, select the files (*index.html and package.json*) and zip them. Now rename the file to *app.nw*. The file structure inside *helloWorld* folder would be



If you are on a Mac, you can navigate back to the *nw* folder and launch the *node-webkit.app* and you will see



click on open and you will see the default welcome page. Close the app. Now if you go back to the *helloWorld* folder, you should see that the OS has recognized the type of the app.



Now, click on *app.nw* and you will see



If you are running on a Windows machine, you need to execute the below command. Make sure you are running the command from inside the *helloWorld* folder

```
../nw/nw.exe app.nw
```

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



As you can see, we have a frame and toolbar on top as it is with any browser. And if you click on the gear icon next to the address bar, the dev tools would open and you can debug the app as you would any other web page.

- Creating a Firebase Powered End to End Ionic Application
- Getting started with MongoDB and Mongoose
- Yeoman, Mongoose and MongoDB

Top5 Posts

- Architecting a Secure RESTful Node.js app - 161,812 views
- Getting Started with Angular 2.0 - 138,237 views
- Developing a MEAN app with Angular 2.0 - 86,985 views
- Node Webkit – Build Desktop Apps with Node and Web Technologies - 67,986 views

Recent Posts

Simple and easy right?

*Note : If you are using Sublime text for your development, you can **check this** to debug the apps without the hassle of zipping and renaming. But for this post, I will be manually doing all the above steps to run the app.*

You can also notice that once you launch the app, it starts behaving like a desktop application. In Mac, you can see the global menu reflecting the current active

← app



Some more config options

node webkit provides 2 more sets of configurations that can control the app. First we will take a look at the *window* object.

In our hello node webkit app, if you would like to get rid of the address bar, you can set the toolbar property to false. To test this, open up *package.json* and update it as

package.json	JavaScript
<pre> 1 { 2 "name": "helloWorld", 3 "version": "0.0.1", 4 "description": "A Hello Node webkit app ", 5 "main": "index.html", 6 "window": { 7 "toolbar": false </pre>	

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

Buy



```

11 },
12 "author": "",
13 "license": "MIT"
14 }

```

To run the app, we need to follow the same steps we did earlier again. select the files (*index.html* and *package.json*) and create a new zip file. And then rename the zip file to *app.nw*. Just to keep track of old

- 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)
 - LESS (3)
 - SASS (1)
- Database (17)
 - Disk DB (6)
 - Firebase (4)
 - Mong
 - MongoDB Lab (3)
- Deployments (4)
 - Heroku (2)
- Desktop Apps (8)
- DNA Analysis (1)
- ECMAScript 6 (3)
- Electron (atom shell) (1)

changes, I will rename the older *nw* file as v1. The file structure would be



Now, launch *app.nw* and you should see



As you must have seen by now, you can resize the window too. You can control that by setting

package.json	JavaScript
1 ...	
2 "window": {	
3 "toolbar": false,	
4 "resizable" : false	
5 }	
6 ...	

And also, you can notice that the window can be resized to a ridiculously small size. You can control the width/min_width/max_width and height/min_height/max_height like

package.json	JavaScript
1 ...	
2 "window": {	
3 "toolbar": false,	
4 "resizable": false,	
5 "width": 800,	
6 "height": 500,	
7 "min_width": 400,	
8 "min_height": 200,	
9 "max_width": 800,	
10 "max_height": 600	
11 },	
12 ...	

You can also control the position where the app opens when launched. You can open the app in the center or at the position of the mouse. The config for that would

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

Buy

package.json	JavaScript
1 ...	
2 "window": {	
3 "toolbar": false,	
4 "resizable": false,	
5 "width": 800,	
6 "height": 500,	
7 "min_width": 400,	
8 "min_height": 200,	
9 "max_width": 800,	
10 "max_height": 600,	
11 "position": "mouse"	

- [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)
 - [Android](#) (15)
 - [Firefo](#) (1)
 - [iOS](#) (1)
 - [PhoneGap](#) (16)
- [My Books](#) (1)
- [Node](#) (60)
 - [CLI](#) (3)
 - [Expressjs](#) (18)
 - [Genome](#) (1)
 - [Grunt](#) (8)
 - [Koa](#) (1)

```
12     }
13     ...
```

You can explore more options [here](#).

Apart from the window, we can also control webkit options too. You can select if you want to load flash/external plugins or Java applets or enable page cache. The updated **package.json** would look like

```
package.json JavaScript
1  {
2    "name": "helloWorld",
3    "version": "0.0.1",
4    "description": "A Hello Node webkit app ",
5    "main": "index.html",
6    "window": {
7      "toolbar": false,
8      "resizable": false,
9      "width": 800,
10     "height": 500,
11     "min_width": 400,
12     "min_height": 200,
13     "max_width": 800,
14     "max_height": 600,
15     "position": "mouse"
16   },
17   "webkit": {
18     "plugin": false,
19     "java" : false,
20     "page-cache" : false
21   },
22   "scripts": {
23     "test": "echo \"Error: no test specified\" && exit 1"
24   },
25   "author": "",
26   "license": "MIT"
27 }
```

You can compress the files, change the extension from zip to **nw** and launch the app. The above settings will not have an impact on our current “Hello World” app.

Adding Native features

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

Menu bar

Now, we have a fair feeling that our app can function like a desktop app, lets us make it more **desktop-ish**. If you did notice, our hello world app has a native minimize, maximize, un-maximize and close buttons. If you want, you can get rid of those by setting `frame :` `false` in the windows config in **package.json**. This will

- 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
- April 2015
- March 2015
- February 2015
- January 2015
- December 2014
- November 2014
- October 2014
- September 2014



remove the buttons and the title bar. Now, you can write your own components and add event listeners to it. You can find more info on that [here](#).

Now, we will try and add a few menu items to our app. I will add a few random menu items, to showcase the feature.

Open *index.html* in your favorite editor and add the below code

```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello Node Webkit</title>
5   <script type="text/javascript">
6     // menu logic goes here
7
8   </script>
9 </head>
10 <body>
11   <h1>Hello Node Webkit</h1>
12 </body>
13 </html>
```

To build a menu, we will be using the `nw.gui` module. As you can read from the name, it holds all the node webkit's GUI modules. We will use the `Menu()` to create an empty menu and using `MenuItem()`, we will add sub-menus to it.

We will need to write different code for Mac and Windows, so the Menu bars look native-ish.

Mac : In *index.html*, below the comment "menu logic"

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

Buy



```
index.html
1 //get node webkit GUI
2   var gui = require('nw.gui');
3
4   // get the window object
5   var win = gui.Window.get();
6
7   var menubar = new gui.Menu({
8     type: 'menubar'
9   });
10
11   var file = new gui.Menu();
```

- [August 2014](#)
- [July 2014](#)
- [June 2014](#)
- [May 2014](#)
- [April 2014](#)
- [March 2014](#)
- [February 2014](#)

Inform me about new posts

Email *

SUBSCRIBE!




```

12     file.append(new gui.MenuItem({
13         label: 'Action 1',
14         click : function () {
15             alert('Action 1 Clicked');
16         }
17     }));
18
19     file.append(new gui.MenuItem({
20         label: 'Action 2',
21         click : function () {
22             alert('Action 2 Clicked');
23         }
24     }));
25
26     // add the menubar to window/menu panel
27     win.menu = menubar;
28     // add the file menu to window/menu panel - menu
29     win.menu.insert(new gui.MenuItem({
30         label: 'File',
31         submenu: file
32     }), 1);

```

Windows : In *index.html*, below the comment “menu logic goes here”, add the code

index.html	JavaScript
1	//get node webkit GUI
2	var gui = require('nw.gui');
3	
4	// get the window object
5	var win = gui.Window.get();
6	var menubar = new gui.Menu({
7	type: 'menubar'
8	});
9	
10	var file = new gui.Menu();
11	file.append(new gui.MenuItem({
12	label: 'Action 1',
13	click: function() {
14	alert('Action 1 Clicked');
15	}
16	});
17	
18	file.append(new gui.MenuItem({
19	label: 'Action 2',
20	click: function() {
21	alert('Action 2 Clicked');
22	}
23	});
24	
25	menubar.append(new gui.MenuItem({ label: 'File', subme

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

Buy



Next, to enable debugging, open *package.json* and set `{toolbar : true}`. Save all the files, zip, rename and launch the app.

On Mac,



And when you click on the menu links, you will see the alerts.

On Windows



As with any menu, you can add sub menus too.

Mac : Update the script in *index.html* to

```
index.html
1 //get node webkit GUI
2   var gui = require('nw.gui');
3
4   // get the window object
5   var win = gui.Window.get();
6
7   var menubar = new gui.Menu({
8     type: 'menubar'
9   });
10
11   var file = new gui.Menu();
12   var subMenu = new gui.Menu();
13
14   file.append(new gui.MenuItem({
15     label: 'Action 1',
16     click: function() {
17       alert('Action 1 Clicked');
18     }
19   }));
20
21   file.append(new gui.MenuItem({
22     label: 'Action 2',
23     click: function() {
24       alert('Action 2 Clicked');
25     }
26   }));
27
28
29   subMenu.append(new gui.MenuItem({
30     label: 'SubMenu Action 1',
31     click: function() {
32       alert('SubMenu Action 1 Clicked');
33     }
34   }));
```

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

Buy



```
37   subMenu.append(new gui.MenuItem({
38     label: 'SubMenu Action 2',
39     click: function() {
40       alert('SubMenu Action 2 Clicked');
41     }
42   }));
43
44   file.insert(new gui.MenuItem({
45     label: 'Sub Action Menu',
46     submenu: subMenu
47   }));
48
49   // add the menubar to window/menu panel
50   win.menu = menubar;
```



```

51 // add the file menu to window/menu panel - menu
52 win.menu.insert(new gui.MenuItem({
53     label: 'File',
54     submenu: file
55 }, 1);

```

Windows : Update the script in *index.html* to

index, .html	JavaScript
1	//get node webkit GUI
2	var gui = require('nw.gui');
3	
4	// get the window object
5	var win = gui.Window.get();
6	var menubar = new gui.Menu({
7	type: 'menubar'
8	});
9	
10	var file = new gui.Menu();
11	var subMenu = new gui.Menu();
12	
13	file.append(new gui.MenuItem({
14	label: 'Action 1',
15	click: function() {
16	alert('Action 1 Clicked');
17	}
18	});
19	
20	file.append(new gui.MenuItem({
21	label: 'Action 2',
22	click: function() {
23	alert('Action 2 Clicked');
24	}
25	});
26	
27	subMenu.append(new gui.MenuItem({
28	label: 'SubMenu Action 1',
29	click: function() {
30	alert('SubMenu Action 1 Clicked');
31	}
32	});
33	
34	subMenu.append(new gui.MenuItem({
35	label: 'SubMenu Action 2',
36	click: function() {
37	alert('SubMenu Action 2 Clicked');
38	}
39	});
40	
41	menubar.append(new gui.MenuItem({ label: 'File', submenu: file
42	file.append(new gui.MenuItem({ label: 'Sub Action Menu
43	

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

Buy



And when you save and launch the app, you should see



on Mac and on Windows



If you are facing issues, you can throw in a

`debugger;` command in the js code. And when the app

is running, you can click on the gear icon and inspect the same in dev tools.



Simple right? You can check out [Menu](#) and [Window Menu](#) docs to get more information.

Tray

We can make our app behave like a “Tray” app (on Mac)



You can update the script in *index.html* with (*both Mac and Windows*)

```
index.html
1  ...
2  // Create a tray icon
3      var tray = new gui.Tray({
4          icon : 'icon.png',
5          title: 'App Tray'
6      });
7
8      // Give it a menu
9      var menu = new gui.Menu();
10     menu.append(new gui.MenuItem({
11         type: 'checkbox',
12         label: 'Are you sure?'
13     }));
14     tray.menu = menu;
15  ...
```

You can read more about Tray apps [here](#).



Clipboard

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

Buy



You can get the contents of the clipboard using the `Clipboard.get()`. Add the below code to *index.html* after the tray setup.

```
index.html
1  ...
2  // Clipboard
3  // We can not create a clipboard, we have to receive the sy
4  var clipboard = gui.Clipboard.get();
5
```

```
6 // Read from clipboard
7 var text = clipboard.get('text');
8 alert('Clipboard text : '+text);
9 ...
```

And when you zip and launch the app, you should see the clipboard text alerted.



You can read more about [platform services](#) here.

Using Node Modules

← We can also leverage the power of the existing node modules to build desktop apps. For example, if you want your app to have a local persistence, you can use a node module named [diskDB](#). This will create a local JSON file which you can manage with a MongoDB like API. →

Let us build a sample settings page using diskDB and node webkit. Create a new folder inside *myWebKitApps* and name it *diskDBApp*. Open terminal/prompt here and run

```
npm init
```

Next, run

```
npm install diskdb --save
```



This will download the diskdb and required files. The

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

Buy



```
package.json
1 {
2   "name": "diskDBApp",
3   "version": "0.0.1",
4   "description": "A diskDB Node webkit app ",
5   "main": "index.html",
6   "window": {
7     "title": "diskDB App",
8     "toolbar": true,
9     "resizable": false,
10    "width": 800,
11    "height": 500,
```

```

12     "min_width": 400,
13     "min_height": 200,
14     "max_width": 800,
15     "max_height": 600,
16     "position": "mouse"
17 },
18 "webkit": {
19     "plugin": false,
20     "java": false,
21     "page-cache": false
22 },
23 "scripts": {
24     "test": "echo \"Error: no test specified\" && exit 1"
25 },
26 "author": "",
27 "license": "MIT",
28 "dependencies": {
29     "diskdb": "^0.1.12"
30 }
31 }

```

← Next, create a new folder named *appPersistence* at the root of the *diskDBApp* folder, where we store the JSON files locally. →

Now, we create *index.html* file at the root of *diskDBApp* and open it in your favorite editor. Update the file like

```

index.html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Hello Node Webkit</title>
6  </head>
7
8  <body>
9      <h1>Persisted Settings</h1>
10     <ul id="settings">
11     </ul>
12     <script type="text/javascript">
13         //get node webkit GUI
14         var gui = require('nw.gui');
15
16         // get the window object
17         var win = gui.Window.get();

```

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

Buy



```

20     var db = require('diskdb');
21     db.connect('appPersistence', ['settings']);
22
23     var _settings = [{
24         "option 1": true
25     }, {
26         "option 2": false
27     }, {
28         "default value": 'foo'
29     }];
30
31     // need not save everytime
32     if (db.settings.count() == 0) db.settings.save(_settings);
33

```



```

34 // get the settings
35 var settings = db.settings.find(),
36     html = '';
37 for (var i = 0; i < settings.length; i++) {
38     var _o = settings[i],
39     li = '<li>';
40     for (var prop in _o) {
41         if (prop != '_id') {
42             li += prop + ' : ' + _o[prop];
43         }
44     }
45     li += '</li>';
46     html += li;
47 };
48 document.getElementById('settings').innerHTML = html;
49 </script>
50 </body>
51
52 </html>

```

← The above code uses **diskDB API** to create a settings file and store the data. Then we will use the `find()` to return all the settings and display the data. Simple and easy right?



To run the app, zip **all** the folders and files and rename the app as **diskDBApp.nw**. And when you launch the app, you will see



Very powerful right? You can use other node package as you need them.

Note : Not all node modules work well on all platforms.

Distribute the App



So far we have zipped and ran the app locally. For the

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

Buy



executable available (**duh!**). This is one of the ways of distributing the app, in case the end users of the app are familiar with node webkit.

Another way of distributing the app is to share your app with the **nw** executable. You can read more about that [here](#).

Another option is to use node-webkit-builder. I found **node-webkit-builder** more reliable while generating cross platform executable. We will be leveraging **Gulp** to perform this task for us. At the root of the **diskDBApp**, create a new file named **GulpFile.js** and add the below contents.

```
gulpFile.js JavaScript
1 var NwBuilder = require('node-webkit-builder');
2 var gulp = require('gulp');
3 var gutil = require('gulp-util');
4
5 gulp.task('nw', function () {
6
7     var nw = new NwBuilder({
8         version: '0.9.2',
9         files: [ './**' ],
10        platforms: ['osx'] // change this to 'win' for/on
11    });
12
13    // Log stuff you want
14    nw.on('log', function (msg) {
15        gutil.log('node-webkit-builder', msg);
16    });
17
18    // Build returns a promise, return it so the task isn't
19    return nw.build().catch(function (err) {
20        gutil.log('node-webkit-builder', err);
21    });
22 });
```

Next, we will install the required dependencies. Run

```
npm install --save-dev gulp gulp-util node-webkit-builder
```

Now run

```
gulp nw
```

This will launch the build task and generate the

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

Buy



After the task is completed, you will see a new folder named **build**, which will consist of the installer

Do notice the size of the app (~90 MB).

If the build is not working as expected on Windows, you can try [nodebob](#).

First, you need to clone the nodebob repo. `CD` into *myNodeWebkitApps* folder and run

```
git clone https://github.com/geo8bit/nodebob.git nodebob
```

Next, navigate to *myNodeWebkitApps\nodebob\app* folder and delete all its contents. Now, copy only the below from *myNodeWebkitApps\diskDBApp* folder

- appPersistence folder
- node_modules folder (*with only diskDB inside it*)
- index.html
- package.json

into *myNodeWebkitApps\nodebob\app* folder.

Back to nodebob folder, Execute (*double click*) on *build.bat* and in a few seconds, the build would be completed. You can navigate to *myNodeWebkitApps\nodebob\release* folder to see the installer generated. It will be named as *nw.exe*.

You can launch it and it should display the setting from the JSON file.



Ann Security

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

Buy



As you already know, we are distributing a zip file that consists of our logic and assets. There may be times, when you would like to protect your app's logic and assets. You can achieve the same using V8 snapshots. More info [here](#).

Hope you got a decent idea on how to build Desktop apps using Node Webkit.

Thanks for reading! Do comment.
@arvindr21

Series Navigation

[Node Webkit powered Dashboard application >>](#)

0 18 0

← Like this:



2 bloggers like this.

Related

[Ionic, Twilio and Node Scheduler - A Reminder App - Part I \[Server\]](#)
September 30, 2014
In "Android"

[Responsive Web Design Test Automation](#)
February 22, 2014
In "Bootstrap"

[Nodejs Restify MongoDB - Build your own REST API](#)
April 20, 2014
In "Database"

JULY 12, 2014

ARVIND RAVULAVARU



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

[Buy](#)



87 Comments

The Jackal of Javascript

Login

Recommend 3

Share

Sort by Best



Join the discussion...

Madhawa Umanga Kariyawasam • 2 years ago

thank you.. it's very useful for developers !

3 ^ | v • Reply • Share ›



feenan • 2 years ago

it's very good, tks!

2 ^ | v • Reply • Share ›

Mad Scientist • 2 years ago

Well written. Thank you.

1 ^ | v • Reply • Share ›

Esneyder Amin. • 2 years ago

gracias fue muy informativo

1 ^ | v • Reply • Share ›



Pim • 2 years ago

Great and clear article, thank you!

1 ^ | v • Reply • Share ›

Nebb • 2 days ago

I have issue on windows build. diskdb seems not to be able to save on settings.json? How can we fix this?

^ | v • Reply • Share ›

mosby josh • 5 days ago

thank you, amazing tutorial

^ | v • Reply • Share ›

Eliyanan • 8 months ago

hello guys, i am getting an error from my ide that nw.gui module is not installed how do i solve it?? please help

^ | v • Reply • Share ›

Ethan Arrowood • a year ago

Hi! I know this was created a while ago, but hopefully you will be able to respond.

I am on a windows pc and I have successfully created an app along with making it single file executable. I want to make my app able to

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

Buy

X

console response you are. It doesn't create a build folder. Let me know if I should post any error messages or what not.

^ | v • Reply • Share ›

Arvind Ravulavaru Mod ➔ Ethan Arrowood • a year ago

What is the error message?

^ | v • Reply • Share ›

Ethan Arrowood ➔ Arvind Ravulavaru • a year ago

... I think the error is related to the ...

node-webkit-builder ERROR: The version 0.9.2 does not have a corresponding build posted at <http://dl.nwjs.io/>. Please choose a version from that list.

node-webkit-builder Unable to download nodewebkit
Both the cache and the node_modules folders are created.


build folder is not created.

^ | v • Reply • Share ›

Arvind Ravulavaru Mod ➔ Ethan Arrowood | 
• a year ago

Replace the version in gulp file from 0.9.2 to 0.13.0. This should fix the above issue.

^ | v • Reply • Share ›

Ethan Arrowood ➔ Arvind Ravulavaru — | 
• a year ago

Didn't work :(

[20:45:06] Using gulpfile

~\Desktop\myNodeWebkitApps\passwordGer

[20:45:06] Starting 'nw'...

[20:45:07] node-webkit-builder Using v0.13.0

[20:45:07] node-webkit-builder Create cache folder in

C:\Users\Ethan\Desktop\myNodeWebkitApps'

[20:45:07] node-webkit-builder Downloading:

<http://dl.nwjs.io/v0.13.0/nwjs...>

[20:45:07] node-webkit-builder Create cache folder in

C:\Users\Ethan\Desktop\myNodeWebkitApps'

[20:45:07] node-webkit-builder Downloading:

<http://dl.nwjs.io/v0.13.0/nwjs...>

[20:45:07] node-webkit-builder ERROR: The version 0.13.0 does not have a corresponding build posted at <http://dl.nwjs.io/>. Please choose a version from that list.

[20:45:07] node-webkit-builder Unable to

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

Buy

X

[20:45:07] node-webkit-builder ERROR: The version 0.13.0 does not have a corresponding build posted at <http://dl.nwjs.io/>. Please choose a version from that list.

^ | v • Reply • Share ›

Arvind Ravulavaru Mod ➔ Ethan Arrowood | 
• a year ago

Try 0.12.3

^ | v • Reply • Share ›

Ethan Arrowood → Arvind Ravulavaru

• a year ago

No error's in the console this time, but I don't get the same output you did in the example. Instead I get a very long chain of directories and files. Is it because I'm on windows and this is for osx?

Here is the comman line info:

```
[20:51:06] Using gulpfile
~\Desktop\myNodeWebkitApps\passwordGer
[20:51:06] Starting 'nw'...
[20:51:07] node-webkit-builder Using v0.12.3
[20:51:07] node-webkit-builder Create cache
folder in
C:\Users\Ethan\Desktop\myNodeWebkitApps'
[20:51:07] node-webkit-builder Downloading:
http://dl.nwjs.io/v0.12.3/nwjs...
[20:51:07] node-webkit-builder Create cache
folder in
C:\Users\Ethan\Desktop\myNodeWebkitApps'
[20:51:07] node-webkit-builder Downloading:
http://dl.nwjs.io/v0.12.3/nwjs...
downloading [=====]
100% 0.0s
```

```
[20:52:22] node-webkit-builder Create release
folder in
C:\Users\Ethan\Desktop\myNodeWebkitApps'
[20:52:22] node-webkit-builder Create release
folder in
C:\Users\Ethan\Desktop\myNodeWebkitApps'
[20:52:47] Finished 'nw' after 1.67 min
```

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Ethan Arrowood

• a year ago

Please go through this for more info:

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

Buy

X

^ | v • Reply • Share ›

Ethan Arrowood → Arvind Ravulavaru

• a year ago

Ok my problem is instead of it creating an executable `node-webkit.app` it is creating a `webkit.projectname.app` directory. Any tips?

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Ethan Arrowood

• a year ago

Take a look at <https://github.com/nwjs/nw-bui...>

^ | v • Reply • Share ›

Ethan Arrowood ➔ Arvind Ravulavaru

• a year ago

I can't figure it out :(Thanks for trying to help me!

^ | v • Reply • Share ›



Ankit • a year ago

Many thanks. This is very useful for first time node web-kit programmers.

^ | v • Reply • Share ›



Vallish Kumaraswami • a year ago

Hi,

We have built a desktop app similar to what you have described above. Now I see a nw.exe running in task manager. But, how do I automate the GUI testing for the same? Are there any tools that can detect the nw.exe type of window to recognize the controls inside the same?

Please help.

^ | v • Reply • Share ›

Cl Sah • a year ago

Hi Arvind,

I use ubuntu and unable to launch node webkit. How to launch node webkit? I don't find any [node-webkit.app](#) after unzip [nwjs-v0.13.0-alpha2-osx-x64.zi...](#) file.

^ | v • Reply • Share ›

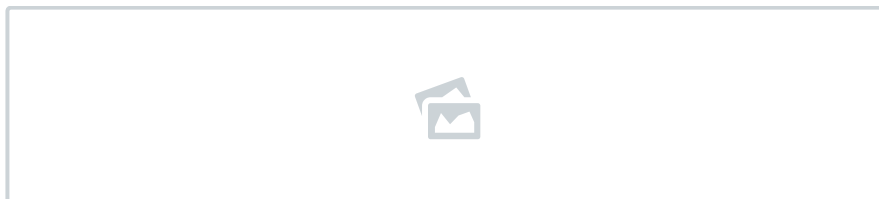
Puneet gupta • a year ago

Hi Arvind,

I tried to make web as desktop app using node-webkit

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

Buy

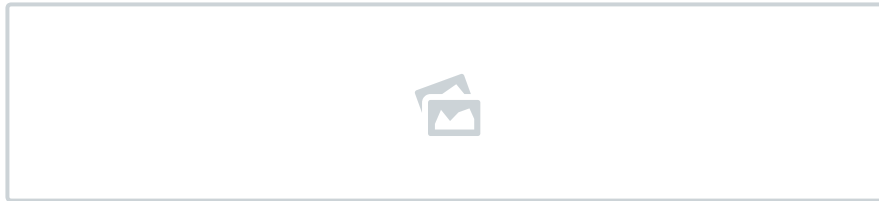


^ | v • Reply • Share ›

Maria Melton • 2 years ago

I'm having a problem with the menu: when I tried it a week ago it worked fine, now when I copied and pasted the code into index.html

I get this error. I'm not sure what it means. I attached the terminal output in the photo below.



^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Maria Melton • 2 years ago — | 🚩

Can you try downloading the OSX libs again? May be they have been updated.

^ | v • Reply • Share ›

← **Maria Melton** • 2 years ago — | 🚩 →

I'm using OSX, this is not working, I followed the directions and when I click on app.nw it just opens the default welcome page again. When I double click index.html I get a new tab in my browser with the desired webpage. What is going on?

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Maria Melton • 2 years ago — | 🚩

Hey Maria, Once you have downloaded the libs can you copy the **nw.app** from the libs folder to the Applications folder of Mac and then try?

^ | v • Reply • Share ›

Harish Jambhale → Arvind Ravulavaru • a year ago — | 🚩

I am having the same issue. I tried copying **nwjs.app** to applications folder, and still the issue persists.

^ | v • Reply • Share ›

Pudjo Isnanto • 2 years ago — | 🚩

can it deployed on web?

^ | v • Reply • Share ›

Maria Melton → Pudjo Isnanto • 2 years ago — | 🚩

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

Buy

X

node webkit wasn't recognized, and even then an alias had to be created to access it. Thanks for getting back to me!

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Pudjo Isnanto • 2 years ago — | 🚩

What do you mean?

^ | v • Reply • Share ›

Maria Melton → Arvind Ravulavaru • 2 years ago — | 🚩

Yes, I figured it out, I wasn't using npm to get node

webkit so node webkit wasn't recognized, and even then an alias had to be created to access it. Thanks for getting back to me!!!

^ | v • Reply • Share ›



Aragorn • 2 years ago

— | 🚩

Thank you, this should get me started for sure.

I have a noob question, does this diskdb a temporary persistence. Like while the app is running data is available and lost when app is closed?

When I tried diskDBApp, json file got created in ../../Temp folder in Windows. I tested by making some changes to the save part, app was not finding the data upon re-launch. I could not figure out how to save the app data permanently.

^ | v • Reply • Share ›

Arvind Ravulavaru Mod ➔ Aragorn • 2 years ago

— | 🚩

Hello Aragorn. Thanks. DiskDB is an JSON based file storage that I have written to manage simple data storage requirements.

To get an understanding of DiskDB, check out :

<http://thejackalofjavascript.c...>

Thanks.

^ | v • Reply • Share ›

Truthers • 2 years ago

— | 🚩

Can I send you my file?

^ | v • Reply • Share ›

Truthers • 2 years ago

— | 🚩

OK - I have loaded my very basic app - into the " Hello Node" index file. And tried to launch it - do I need to change anything in the json file?

^ | v • Reply • Share ›

Arvind Ravulavaru Mod ➔ Truthers • 2 years ago

— | 🚩

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

Buy

X

^ | v • Reply • Share ›

Truthers • 2 years ago

— | 🚩

Yes it did - thanks a lot!

^ | v • Reply • Share ›

Truthers • 2 years ago

— | 🚩

Thanks - Hate you for being so javascript-cute!!

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Truthers • 2 years ago

No problem. Did the solution work?

^ | v • Reply • Share ›

Truthers • 2 years ago

Hi Arvind - OK. Can you take me through it:

1. I have Yosemite

2. I have node -v 12.0

3. I have newest node webkit : Mac 10.7 + /64bit

I create a "hello world" folder, as well as a nw. folder

I created the json file exactly as you said.

I created the index.html file exactly as copied from this site.

Then in that same folder (the Hello world folder- I compressed the Index and json file and renamed it app.nw.

I then took the node webkit folder (after I unzipped it) and pasted into the nw folder

By now nothing is like you say on your site - the app.nw recognises itself before I open the node-webkit (or now the nwjs file)

What am I doing wrong - it should not be as difficult as this -

PLEASE HELP.

Regards

Mac 10.7+: 32bit / 64bit

^ | v • Reply • Share ›

Arvind Ravulavaru Mod → Truthers • 2 years ago

Hello Liesl, Follow the below.

1. Unzip the node-webkit zip file and copy the **nwjs.app** to your applications folder

2. Double click the app.nw file you have created.

This should open your app using the **nwjs.app** from applications folder.

^ | v • Reply • Share ›

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

Buy

X

Does not even work if you use the Sublime method. Can someone please update this - there is some serious issues with having this work on Yosemite - Newest Version Node - Newest Version Webkit

^ | v • Reply • Share ›

Truthers • 2 years ago

Hi I have a problem making the app launch. I am using Yosemite -

Node.js 10.22 and nodewebkit. You are giving the solution down

here by using this: Step 1 : Open sublime text, next go to Tools build

system and New build system

Step 2 : Copy paste the below code inside it and save it in the default location as node-webkit.sublime-build

```
{  
  "cmd": ["node-webkit", "--enable-logging",  
    "${project_path}:${file_path}"],  
  "working_dir": "${project_path}:${file_path}",  
  "path": "/Applications/node-webkit.app/Contents/MacOS..."  
}
```

The default path would be

"\${home}/\${username}/\${library}/Application Support/Sublime Text



ALL RIGHTS RESERVED | THE JACKAL OF JAVASCRIPT



⌂



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

Buy

