

GETTING STARTED WITH NODE-WEBKIT

by Kristi Centinaro | [@theresnobox](#)

WHAT IS NODE-WEBKIT?

It's a way to create native desktop applications using web technologies (HTML, CSS, JavaScript) by leveraging the power of NodeJS and the Webkit engine.

PREREQUISITES

A basic understanding of these things will make using Node-Webkit way easier.

- A basic understanding of NodeJS
 - how require and module.exports function
 - how to use and install node packages with `npm install`
- Node and NPM installed on your development machine and available in the command line
- Comfort using the command line to install packages and execute tasks

DEVELOPING WITH NODE-WEBKIT: BINARIES

Two ways to install binaries:

THE HARD WAY:

Download from Github Repo & configure yourself

THE EASY WAY:

Use the convenient `nodewebkit` node package

ANATOMY OF A NODE-WEBKIT APP

index.html

package.json

Everything else is up to you!

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

THE HELLO WORLD APP

HTML:

```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

PACKAGE.JSON:

```
{  
  "name": "hello-world-app",  
  "version": "0.0.1",  
  "main": "./index.html",  
  "scripts": {  
    "start": "nodewebkit"  
  },  
  "devDependencies": {  
    "nodewebkit": "^0.10.2"  
  }  
}
```


THE TODO LIST

FILES SETUP

```
MyAwesomeApp
|- app/
|   |- node_modules/
|   |- package.json
|   |- index.html
|- package.json
|- node_modules/
|- .gitignore
```

ADDING AN APP MENU

This is an app menu that binds to the todo's create event:

```
// link to the gui api
var gui = require('nw.gui');

// create an instance of the menu
var menu = new gui.Menu();

// add a menu item that triggers a new todo on click
menu.append(new gui.MenuItem({
  label: 'New ToDo Item',
  click: function() {
    app.todos.create({
      title: 'New ToDo Item',
      order: app.todos.nextOrder(),
      completed: false
    });
  }
}));
```

CHANGING THE APP WINDOW

You can adjust the dimensions and default display settings.

```
// package.json
"window": {
  "title": "My ToDo List",
  "toolbar": false,
  "width": 800,
  "height": 500
}
```

PERSISTING (SAVING) APP DATA

A lot of methods are supported by default

- Web SQL
- Embedded Databases
- Web Storage (LocalStorage)
- IndexedDB

PACKAGING FOR DISTRIBUTION

THE HARD WAY:

Configure and compile yourself for each OS

THE EASY WAY:

Use the convenient `node-webkit-builder` package!

Bonus: Use `grunt-node-webkit-builder` and configure deployments once. Woot.

GRUNT NODE-WEBKIT BUILDER

```
// Project configuration.
grunt.initConfig({
  nodewebkit: {
    options: {
      // OS's to build for
      platforms: ['win','osx'],

      // Where the build version of
      // my node-webkit app is saved
      buildDir: './webkitbuilds',
    },
    // Your node-webkit app
    src: ['./app/**/*']
  },
});

grunt.loadNpmTasks('grunt-node-webkit-builder');
```

THANKS!

SLIDES & FILES AT:

bit.ly/nw-intro

KRISTI CENTINARO

kristi.centinaro.net

[@theresnobox](#)

OTHER USEFUL RESOURCES

- [Creating Desktop Applications with Node-Webkit](#)
- [Getting Started with Node-Webkit](#)
- [Persisting Data in Node-Webkit](#)
 - [IndexedDB Api](#)
 - [About PouchDB](#)
- [How to Package and Distribute Your Apps](#)
 - [Grunt Node-Webkit Builder](#)