

WEBSITE WRAPPER

Overview

The code is built with the core job of taking existing web apps and wrapping them into a native application for OSX or Windows. Is based on the micro services pattern. Where they work independent of each other.

Github Repo: <https://github.com/RobertJGabriel/websiteWrapper>

Npm Location: <https://www.npmjs.com/package/websitewrapper>

Key Features

1. Builds a native app for OS X automatically.
2. Builds a native exe for Windows automatically.
3. Allows for autobuilding of applications
4. Built to work in command and in Node/Gulp.js file.

Installation

Open up cmd/terminal and type the following
`npm install websitewrapper -g`

Usage

You need to pass three things into it.

`-u` or `-url` : The url of the website/web app you want to create the shortcut for.

`-t` or `-title` : The title of the website.

`-i` or `icon`: The icon (its full path) you want pass (To convert images to icns visit [iconverticons](#))

It will then create a folder called build on your desktop. Everything is in there now.

Examples

Command

```
websitewrapper -url http://www.google.ie -title Google -icon /Users/robertjgabriel/desktop/icon.icn
```

Code

```
var websitewrapper = require('websitewrapper'),
    websitewrapper.create("http://www.google.ie", "google", "logo.icns");
```

Built with

Its built used the following packages.

1. **Fs:** Create and delete files in node
2. **nw-builder:** Lets you call all Node.js modules directly from DOM and enables a new way of writing applications with all Web technologies. <https://github.com/nwjs/nw-builder>
3. **file-move:** This is a module to move file <https://www.npmjs.com/package/file-move>
4. **gulp-rename:** gulp-rename is a gulp plugin to rename files easily. : <https://www.npmjs.com/package/gulp-rename>
5. **gulp:** gulp is a toolkit that helps you automate painful or time-consuming tasks in your development workflow. <https://www.npmjs.com/package/gulp>
6. **osenv:** Look up environment settings specific to different operating systems. <https://www.npmjs.com/package/osenv>
7. **chalk:** Terminal string styling done right. Much color. <https://www.npmjs.com/package/chalk>
8. **username:** Get the username of the current user. <https://www.npmjs.com/package/username>

The way the code works is that the user passes in 3 parameters either by using required and the built in create function or though the command line.

How the code works

This information that is passed is then injected into a html template

```
var html = '<!DOCTYPE html>' +
  '<html lang="en">' +
  '<head style="border:0px;margin:0px;">' +
  '<title>' + title + '</title>' +
  '<script>' +
  'var gui=require("nw.gui"),menu=new gui.Menu({type:"menubar"}),menuItems=new
gui.Menu;menu.createMacBuiltin("'" + title + "',{hideEdit:!1,hideWindow:!
0}),gui.Window.get().menu=menu;' +
  '</script>' +
  '</head>' +
  '<body style="border:0px;margin:0px;">' +
  '<iframe src="' + url + '" style="border:0;width:100vw;height:100vw;padding:0;"></iframe>' +
  '</body>' +
  '</html>';
```

Which is then saved into a folder called temp under the file name index.html. This temp folder is used to build the app, so we also store the default package file settings for node and the icon.

Then nw.js files moves these files along with the icon that the user passed in and compiles it into a folder on the users desktop. This was done with the username module in node. The reason for this , was cause if the user was in sudo node it appears as root and not as the login user. This module solves that.

The code also does checks for correct permissions and file handling on both windows and osx.

The use of the gulp-rename renames the icon the user passes in to the icon.icns

```
gulp.src-icons)
  .pipe(rename("icon.icns"))
  .pipe(gulp.dest(__dirname + '/temp'));
```

The below is the example for the build settings used for compiling the files into the app or exe programme

```
var nw = new NwBuilder({
  files: [__dirname + '/temp/index.html', __dirname + '/temp/icon.icns', __dirname + '/temp/
    package.json', __dirname + 'node_modules/**'], // use the glob format
  platforms: ['osx64', 'win64'],
  appName: title,
  macIcns: __dirname + "/temp/icon.icns",
  buildDir: buildPath,
  version: "0.12.0",
  zip: false
});
```

- * __dirname = the location of the module
- * buildPath = the full path to the users desktop
- * version = is the chrome api version to use for the node webkit

