Requiring electron outside of main.js causes a TypeError #7300

New issue

Closed nukeop opened this issue on Sep 21, 2016 · 13 comments

nukeop commented on Sep 21, 2016

- Electron version: 1.3.5
- Operating system: Mint 17

Hello

I am using Electron with React.js and babelify, among other plugins (list at the bottom).
Attempting to use `require('electron')`, for example to gain access to electron's `BrowserWindow`, results in the console showing the following error:
`index.js:4 Uncaught TypeError: fs.readFileSync is not a function`

I can, however, use `const electron = require('electron');` in main.js. For what it's worth, I am also using watchify to pack everything into a js bundle.

Here is the complete list of dependencies in my package.json:

```
"devDependencies": {
        "axios": "^0.9.1",
        "babel-preset-es2015": "^6.6.0",
        "babel-preset-react": "^6.5.0",
        "babelify": "^7.2.0",
        "classnames": "^2.2.3",
        "electron": "^1.3.5",
        "electron-reload": "^0.2.0",
        "jquery": "^2.2.3",
        "react": "^0.14.0",
        "react-autocomplete": "^1.0.0-rc2",
        "react-dom": "^0.14.7",
        "react-sound": "^0.4.0",
        "soundmanager2": "^2.97.20150601-a"
    },
```

kevinsawicki commented on Sep 21, 2016 Contributor

> index.js:4 Uncaught TypeError: fs.readFileSync is not a function

Can you include line 4 of `index.js` here? Or a fuller stack trace?

kevinsawicki added the more-information-needed label on Sep 21, 2016

nukeop commented on Sep 21, 2016

I believe it is referring to index.js of electron, which is itself located in node-modules of the project. Here are the contents of this file:

```
var fs = require('fs')
var path = require('path')

module.exports = path.join(__dirname, fs.readFileSync(path.join(__dirname, 'path.txt'), 'u
```

The entire stack trace suggests a conflict with React.js:

```
Uncaught TypeError: fs.readFileSync is not a function                    index.js:4
    (anonymous function)                                                 index.js:4
    29.fs                                                                index.js:5
    s                                                                    _prelude.js:1
    (anonymous function)                                                 _prelude.js:1
    render                                                               navbar.component.js:24
    _renderValidatedCompositeWithoutOwnerOrContext                       ReactCompositeComponent.js:587
    _renderValidatedComponent                                            ReactCompositeComponent.js:607
    ReactCompositeComponent__renderValidatedComponent                    ReactPerf.js:66
    mountComponent                                                       ReactCompositeComponent.js:220
    ReactCompositeComponent_mountComponent                               ReactPerf.js:66
    mountComponent                                                       ReactReconciler.js:37
    mountChildren                                                        ReactMultiChild.js:241
    _createContentMarkup                                                 ReactDOMComponent.js:591
    mountComponent                                                       ReactDOMComponent.js:479
    mountComponent                                                       ReactReconciler.js:37
    mountComponent                                                       ReactCompositeComponent.js:225
    ReactCompositeComponent_mountComponent                               ReactPerf.js:66
    mountComponent                                                       ReactReconciler.js:37
    mountComponent                                                       ReactCompositeComponent.js:225
    ReactCompositeComponent_mountComponent                               ReactPerf.js:66
    mountComponent                                                       ReactReconciler.js:37
    mountComponent                                                       ReactCompositeComponent.js:225
    ReactCompositeComponent_mountComponent                               ReactPerf.js:66
    mountComponent                                                       ReactReconciler.js:37
    mountComponentIntoNode                                               ReactMount.js:266
    perform                                                              Transaction.js:136
    batchedMountComponentIntoNode                                        ReactMount.js:283
    perform                                                              Transaction.js:136
    batchedUpdates                                                       ReactDefaultBatchingStrategy.js:62
    batchedUpdates                                                       ReactUpdates.js:94
    _renderNewRootComponent                                              ReactMount.js:476
    ReactMount__renderNewRootComponent                                   ReactPerf.js:66
    _renderSubtreeIntoContainer                                          ReactMount.js:550
    render                                                               ReactMount.js:570
    React_render                                                         ReactPerf.js:66
    1../containers/app.container                                         app.js:16
    s                                                                    _prelude.js:1
    e                                                                    _prelude.js:1
    (anonymous function)                                                 _prelude.js:1
```

kevinsawicki commented on Sep 21, 2016 Contributor

What happens if you run `require('fs').readFileSync` from the Console tab of the dev tools?

It looks like the `electron-prebuilt` node module (the file you pasted) is ending up in your packaged app which I don't think you want since it should use the built-in `electron` require instead.

MarshallOfSound commented on Sep 21, 2016 Member

@**nukeop** How are you launching your app. Your `start` script should look like.

```
"scripts": {
  "start": "electron ."
}
```

I have a feeling you are attempting to run your `main.js` with node

```
node main.js
```

Which won't work

nukeop commented on Sep 21, 2016

> What happens if you run require('fs').readFileSync from the Console tab of the dev tools?

Right after this error is thrown, I am able to run `require('fs').existsSync` in console to print a definition of the function. It also works before the error.

> It looks like the electron-prebuilt node module (the file you pasted) is ending up in your packaged app which I don't think you want since it should use the built-in electron require instead.

I have a watchify instance running in the background as I'm developing which is continually updating my package. I defined it in scripts section of package.json like this:

`"watch": "watchify app/app.js -t babelify -o public/js/bundle.js --debug --verbose"`

Any advice on avoiding bundling `electron-prebuilt`?

MarshallOfSound commented on Sep 21, 2016 Member

@**nukeop** Electron supports require internally so you don't need to use browserify.

As far as I am aware, I'd you want to use browserify, you have to exclude "electron".

nukeop commented on Sep 21, 2016

Interesting, could it be that watchify/browserify is ruining this? It has worked ok so far.

Now I am not sure how to run the program without it.

MarshallOfSound commented on Sep 21, 2016 Member

Literally just run

```
electron .
```

From your main app folder, you don't need to bundle anything when using Electron as it has a full node environment internally.

I'm going to close this out as it is a Browserify issue

MarshallOfSound closed this on Sep 21, 2016

nukeop commented on Sep 21, 2016

That's what I've been doing all along, packing the program into a single .js file, which is included in a simple html file with:

```
<script src="public/js/bundle.js">
</script>
```

And everything works, except when I use require. This is a problem with interaction between the two modules.

If I do not pack the whole program into a bundle, I have no easy way of running it, as my main.js only starts electron and loads the html file that includes the bundle.

MarshallOfSound commented on Sep 21, 2016 Member

@**nukeop** The renderer process inside Electron has access to a full node/commonjs environment as well, so you don't need to bundle anything.

> If I do not pack the whole program into a bundle, I have no easy way of running it, as my main.js only starts electron and loads the html file that includes the bundle.

I'm not sure I understand here, any script you load in your HTML file has a full commonjs environment and can therefore use `require` to load in extra files without browserifying anything

nukeop commented on Sep 21, 2016

For anyone encountering this problem in the future and reading this thread, using `window.require` instead of require is one possibility of avoiding the conflict between electron's and browserify's `require` function.

6 6 2

srinathh commented on Jan 21

FWIW, I ran into the same issue trying to use electron in the renderer process with create-react-app that uses webpack in the backend instead of browserify. `window.require` seems to solve it for me too though I'm not entirely clear why.

4 1

Grandclosing commented 6 days ago

@**nukeop** - thanks for your last post; it helped me a lot. `window.require` worked for me.

1

Sign up for free to join this conversation on GitHub. Already have an account? Sign in to comment

© 2017 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About