

Multiple Aliases


In this lesson, we'll deal with Multiple aliases.

WE'LL COVER THE FOLLOWING ^

- Other Solutions
- Wrapping Up

Very often, **multiple aliases** are used for convenience, so instead of using just an `@` to define your root folder, you use many. For example, let's say we have an `action` and a `models` folder. If you create an alias for each one, and then you move the folders around, you just need to change the aliases instead of updating all the references to it in the codebase. That's the power of module aliases: they make your codebase cleaner and more maintainable.

Let's add a `components` alias in `build/webpack.base.conf.js`:

 `build/webpack.base.conf.js`

```
module.exports = {
  // ...
  resolve: {
    extensions: [".js", ".vue", ".json"],
    alias: {
      vue$: "vue/dist/vue.esm.js",
      "@": path.join(__dirname, "..", "src"),
      components: path.join(__dirname, "..", "src", "components")
    }
  }
};
```

Then, we just need to add it to the Jest configuration in `package.json`:

```
{
  "jest": {
    "moduleNameMapper": {
      "@(.*)$": "<rootDir>/src/$1",
      "components( *)$": "<rootDir>/src/components/$1"
```

```

    components({})$ : {rootDir}/src/components/$1 ;
    "^vue$": "vue/dist/vue.common.js"
  }
}
}
}

```

As simple as that. Now, we can try to use both forms in `App.vue`:

```

require('./check-versions')()

process.env.NODE_ENV = 'production'

var ora = require('ora')
var rm = require('rimraf')
var path = require('path')
var chalk = require('chalk')
var webpack = require('webpack')
var config = require('../config')
var webpackConfig = require('./webpack.prod.conf')

var spinner = ora('building for production...')
spinner.start()

rm(path.join(config.build.assetsRoot, config.build.assetsSubDirectory), err => {
  if (err) throw err
  webpack(webpackConfig, function (err, stats) {
    spinner.stop()
    if (err) throw err
    process.stdout.write(stats.toString({
      colors: true,
      modules: false,
      children: false,
      chunks: false,
      chunkModules: false
    }) + '\n\n')

    console.log(chalk.cyan(' Build complete.\n'))
    console.log(chalk.yellow(
      ' Tip: built files are meant to be served over an HTTP server.\n' +
      ' Opening index.html over file:// won\'t work.\n'
    ))
  })
})

```

Run the tests and that should work as well as if you run `npm start` and try it.

Other Solutions

You can see [babel-plugin-webpack-alias](#), especially used for other testing frameworks, such as [mocha](#), which don't have a module mapper.

Wrapping Up

Adding module aliases is very simple and can keep your codebase much cleaner and easier to maintain. Jest makes it very easy to define them, you just need to keep it in sync with the Webpack aliases.

Let's test a running project of what we have done so far in the next lesson.