Node, Express

Node

The actual shell your script is run within is platform dependent. By default, on Unix-like systems it is the /bin/sh command, on Windows it is cmd.exe. The actual shell referred to by /bin/sh also depends on the system. You can customize the shell with the script-shell configuration.

# upgrade npm itself

npm install –g npm

# UMD

[UMD (Universal Module Definition](https://github.com/umdjs/umd)) is a block of code we can use to wrap around our library. This block of code makes it possible to use a library both on the frontend and in Node.

It kinda looks like this:

// This is js

(function (root, factory) { if (typeof define === 'function' && define.amd) { // AMD. Register as an anonymous module. define(['b'], factory); } else if (typeof module === 'object' && module.exports) { // Node. module.exports = factory(require('b')); } else { // Browser globals (root is window) root.returnExports = factory(root.b); }}(typeof self !== 'undefined' ? self : this, function (b) { // Use b in some fashion.

// Just return a value to define the module export. // This example returns an object, but the module // can return a function as the exported value. return {};}));

I used the following and it worked for a single file:

# Run Node online

<https://runkit.com/ron2015schmitt/>

# npm install

For npm install specific version, use npm install [package-name]@[version-number].

Use npm view [package-name] version to know the specific latest version of a package available on the npm registry.

Use npm list [package-name] to know the specific latest version of an installed package.

Use npm install [package-name]@[version-number] to install an older version of a package.

Prefix a version number with a **caret** (^) or a **tilde** (~) to specify to install the **latest minor** or **patch** version, respectively.