**What is the purpose of Node.js module.exports and how do you use it?**

*1478*

*down vote*

*accepted*

*module.exports is the object that's actually returned as the result of a require call.*

*The exports variable is initially set to that same object (i.e. it's a shorthand "alias"), so in the module code you would usually write something like this:*

*var myFunc1 = function() { ... };*

*var myFunc2 = function() { ... };*

*exports.myFunc1 = myFunc1;*

*exports.myFunc2 = myFunc2;*

*to export (or "expose") the internally scoped functions myFunc1 and myFunc2.*

*And in the calling code you would use:*

*var m = require('./mymodule');*

*m.myFunc1();*

*where the last line shows how the result of require is (usually) just a plain object whose properties may be accessed.*

*NB: if you overwrite exports then it will no longer refer to module.exports. So if you wish to assign a new object (or a function reference) to exports then you should also assign that new object to module.exports*

*It's worth noting that the name added to the exports object does not have to be the same as the module's internally scoped name for the value that you're adding, so you could have:*

*var myVeryLongInternalName = function() { ... };*

*exports.shortName = myVeryLongInternalName;*

*// add other objects, functions, as required*

*followed by:*

*var m = require('./mymodule');*

*m.shortName(); //*