



# REQUIRING MODULES



```
var http = require('http');
```

-----➔ http.js

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

-----➔ fs.js

How does 'require' return the libraries?

How does it find these files?

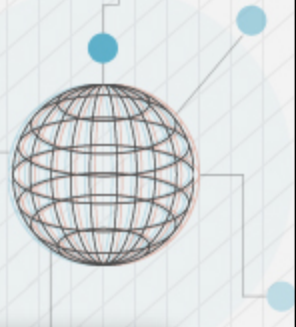


MODULES





# LETS CREATE OUR OWN MODULE



## custom\_hello.js

```
var hello = function() {  
  console.log("hello!");  
}  
exports = hello;
```

## custom\_goodbye.js

```
exports.goodbye = function() {  
  console.log("bye!");  
}
```

## app.js

*exports defines what require returns*

```
var hello = require('./custom_hello');  
var gb = require('./custom_goodbye');  
hello();  
gb.goodbye();
```

```
require('./custom_goodbye').goodbye();
```

*If we only need to call once*



MODULES





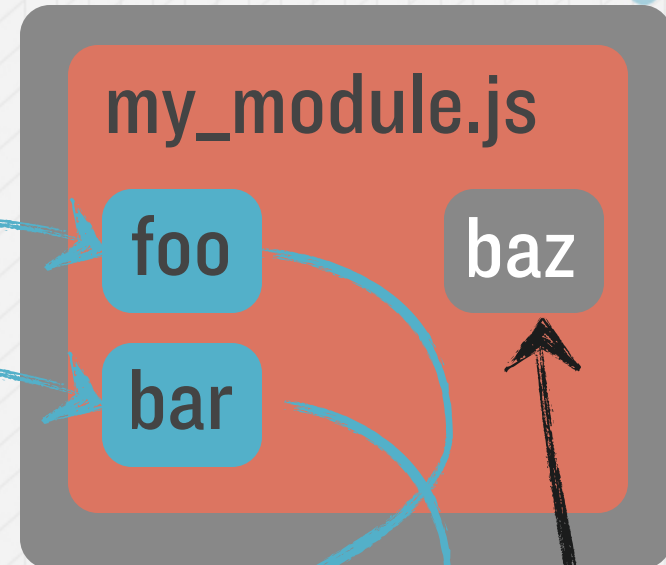
# EXPORT MULTIPLE FUNCTIONS



## my\_module.js

```
var foo = function() { ... }  
var bar = function() { ... }  
var baz = function() { ... }
```

```
exports.foo = foo  
exports.bar = bar
```



## app.js

```
var myMod = require('./my_module');  
myMod.foo();  
myMod.bar();
```

*"private"*



MODULES





# MAKING HTTP REQUESTS

app.js

```
var http = require('http');

var message = "Here's looking at you, kid.";
var options = {
  host: 'localhost', port: 8080, path: '/', method: 'POST'
}

var request = http.request(options, function(response){
  response.on('data', function(data){
    console.log(data); logs response body
  });
});
request.write(message); begins request
request.end(); finishes request
```



MODULES





# ENCAPSULATING THE FUNCTION



app.js

```
var http = require('http');

var makeRequest = function(message) {
  var options = {
    host: 'localhost', port: 8080, path: '/', method: 'POST'
  }

  var request = http.request(options, function(response){
    response.on('data', function(data){
      console.log(data);
    });
  });
  request.write(message);
  request.end();
}

makeRequest("Here's looking at you, kid.");
```



MODULES







# CREATING & USING A MODULE



```
var http = require('http');
```

make\_request.js

```
var makeRequest = function(message) {  
  ...  
}
```

```
exports = makeRequest;
```

```
var makeRequest = require('./make_request');
```

app.js

```
makeRequest("Here's looking at you, kid");  
makeRequest("Hello, this is dog");
```

## Where does require look for modules?

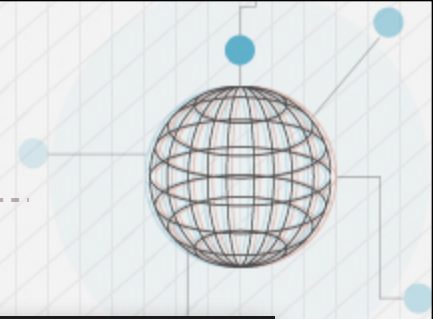


MODULES





# REQUIRE SEARCH



```
var make_request = require('./make_request') look in same directory  
var make_request = require('../make_request') look in parent directory  
var make_request = require('/Users/eric/nodes/make_request')
```

`/Home/eric/my_app/app.js`

*Search in node\_modules directories*

```
var make_request = require('make_request')
```

- `/Home/eric/my_app/node_modules/`
- `/Home/eric/node_modules/make_request.js`
- `/Home/node_modules/make_request.js`
- `/node_modules/make_request.js`

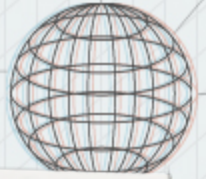


MODULES





# NPM: THE USERLAND SEA



## Package manager for node

- Comes with node
- Module Repository
- Dependency Management
- Easily publish modules
- “Local Only”



“Core” is small. “Userland” is large.



MODULES







# INSTALLING A NPM MODULE



In /Home/my\_app

```
$ npm install request
```

 <https://github.com/mikeal/request>

Installs into local node\_modules directory

Home



my\_app



node\_modules



request

In /Home/my\_app/app.js

```
var request = require('request');
```

Loads from local node\_modules directory



MODULES





# LOCAL VS GLOBAL



Install modules with executables globally

```
$ npm install coffee-script -g
```

*global*

```
$ coffee app.coffee
```

Global npm modules can't be required

```
var coffee = require('coffee-script');
```



```
$ npm install coffee-script
```

*Install them locally*

```
var coffee = require('coffee-script');
```



MODULES





# FINDING MODULES



## npm registry

npm registry

Find packages... or browse

	Latest Updates	Most Depended
adstream-data	7 minutes ago	underscore
grover	7 minutes ago	coffee-script
tint	28 minutes ago	request
genetics	44 minutes ago	express
oauth2-node	49 minutes ago	async
kdtree	54 minutes ago	optimist
stateful	1 hour ago	colors
siq-mesh	1 hour ago	connect

## npm command line

```
$ npm search request
```

## toolbox.no.de

node-toolbox Categories Packages Search

### References

- Official Node.js Documentation
- Felix's Node.js Guide
- The Node Beginner Book
- Up and Running with Node.js
- Mastering Node
- Hands-on Node.js

### Popular by github

- joyentexpress
- express
- d3
- ender-d3
- blueimp-file-upload-node
- blueimp-file-upload-jquery-ui
- blueimp-file-upload
- coffee-script
- socket.io-wisdom

## github search

github Search...

Explore Repositories

Advanced Search



MODULES





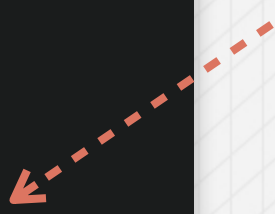
# DEFINING YOUR DEPENDENCIES



my\_app/package.json

```
{  
  "name": "My App",  
  "version": "1",  
  "dependencies": {  
    "connect": "1.8.7"  
  }  
}
```

*version number*



```
$ npm install
```

Installs into the node\_modules directory

my\_app



node\_modules



connect



MODULES







# DEPENDENCIES



my\_app/package.json

```
"dependencies": {  
  "connect": "1.8.7"  
}
```

No conflicting modules!

Installs sub-dependencies

my\_app



node\_modules



connect

connect



node\_modules



qs

connect



node\_modules



mime

connect



node\_modules



formidable



MODULES





# SEMANTIC VERSIONING



"connect": "1.8.7"

*Major*     *Minor*     *Patch*  
1     8     7

## Ranges

"connect": "~1"



$\geq 1.0.0$   $< 2.0.0$

*Dangerous*

"connect": "~1.8"



$\geq 1.8$   $< 2.0.0$

*API could change*

"connect": "~1.8.7"



$\geq 1.8.7$   $< 1.9.0$

*Considered safe*

<http://semver.org/>



MODULES

