



MODULES

- LEVEL FOUR -

.....





REQUIRING MODULES



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

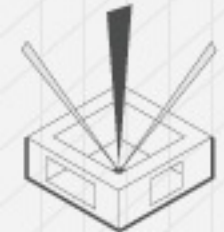
-----➔ http.js

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

-----➔ fs.js

How does 'require' return the libraries?

How does it find these files?

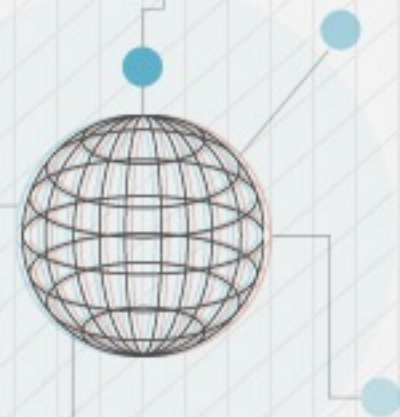


MODULES





LET'S CREATE OUR OWN MODULE



custom_hello.js

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

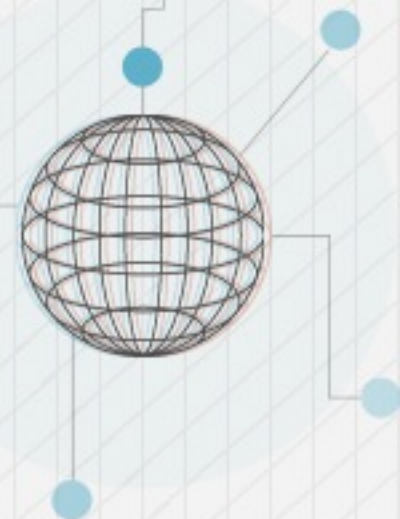
If we only need to call once



MODULES



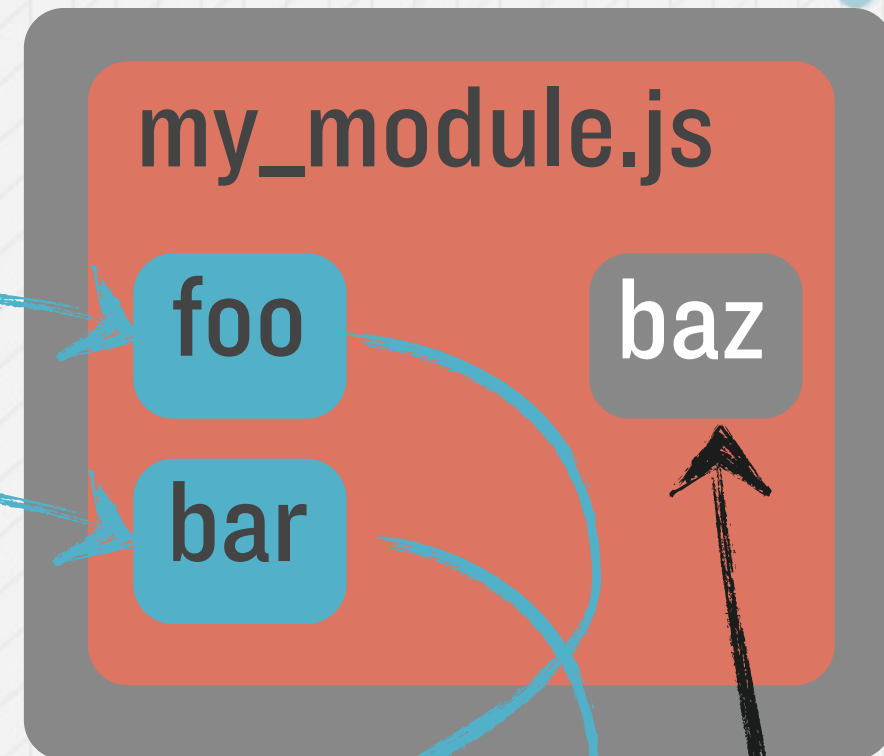
EXPORT MULTIPLE FUNCTIONS



my_module.js

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

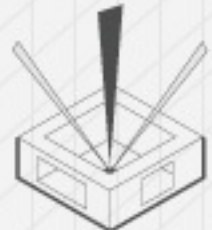
```
exports.foo = foo  
exports.bar = bar
```



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) {  
    ...  
}
```

```
module.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?

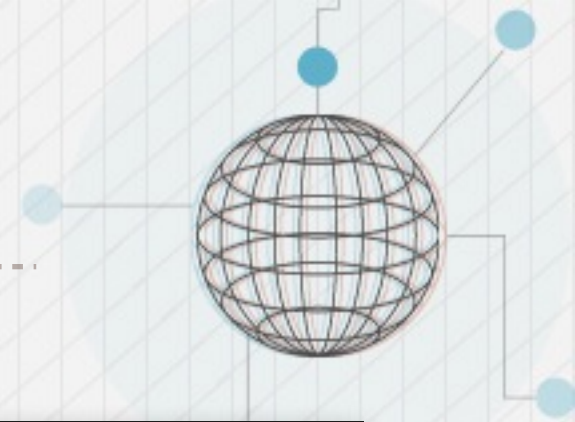


MODULES





REQUIRE SEARCH



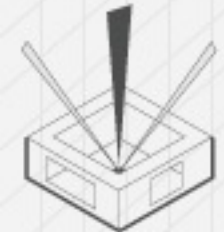
```
var make_request = require('./make_request') look in same directory  
var make_request = require('../make_request') look in parent directory  
var make_request = require('/Users/eric/nodes/make_request')
```

/Home/eric/my_app/app.js

Search in node_modules directories

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

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



MODULES

However, all packages are directories



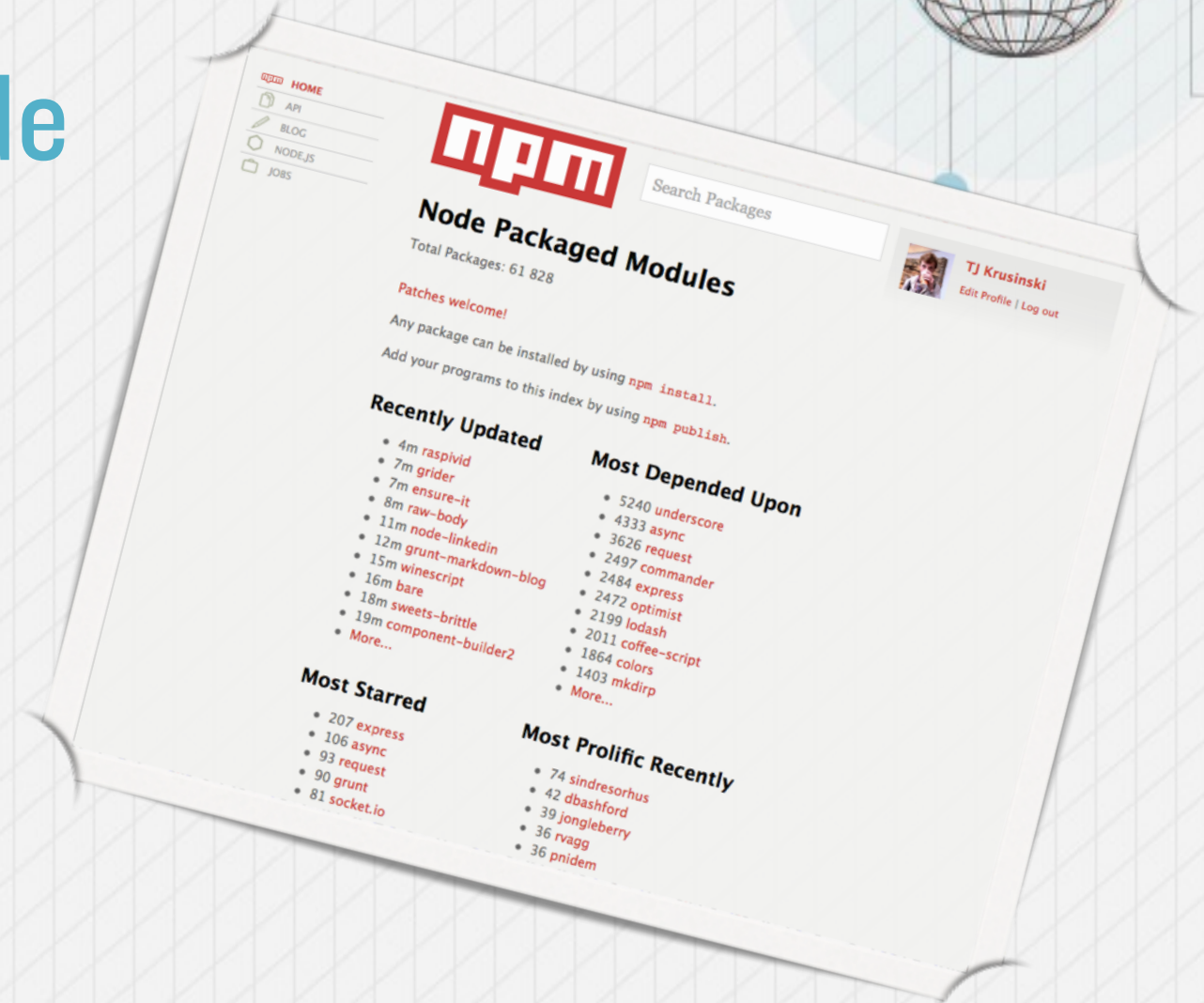


NPM: THE USERLAND SEA



Package manager for node

- Comes with node
- Module Repository
- Dependency Management
- Easily publish modules



<http://npmjs.org>



MODULES





INSTALLING A NPM MODULE



In /Home/my_app

```
$ npm install 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

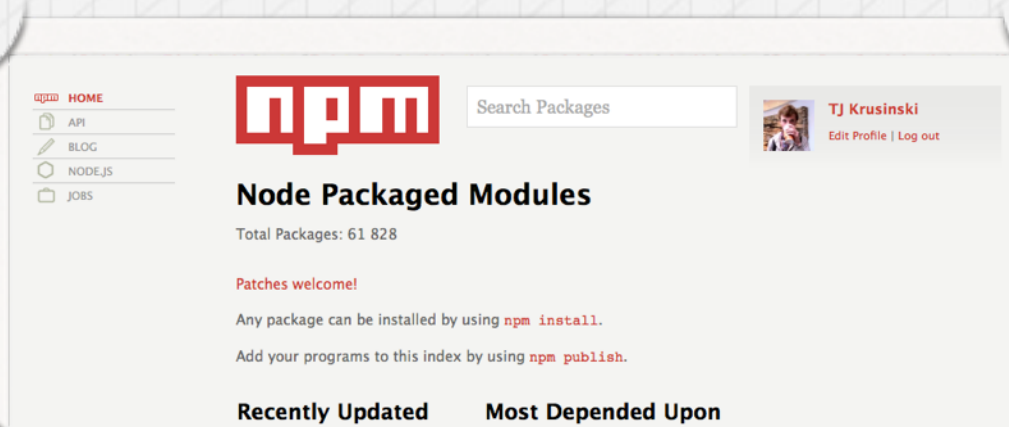




FINDING MODULES



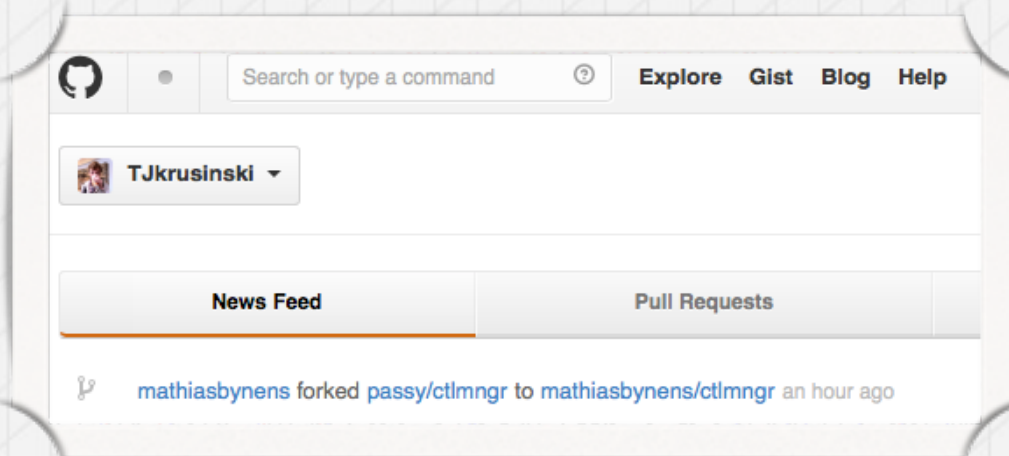
npm registry



npm command line

```
$ npm search request
```

github 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"  
}
```

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"
```



```
>=1.0.0 <2.0.0
```

Dangerous

```
"connect": "~1.8"
```



```
>=1.8.0 <1.9.0
```

API could change

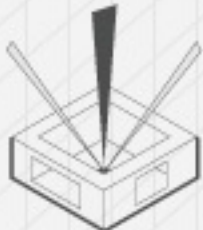
```
"connect": "~1.8.7"
```



```
>=1.8.7 <1.9.0
```

Considered safe

<http://semver.org/>



MODULES

