NodeJS. GLOBAL

NodeJS Modules. NPM. Yarn.

April, 2018

AGENDA

- Standard modules
- Module load systems
- Dive into module object
- How does Node find the module?
- NPM
- NPM package structure
- Create custom package
- Global modules
- Yarn
- Q&A

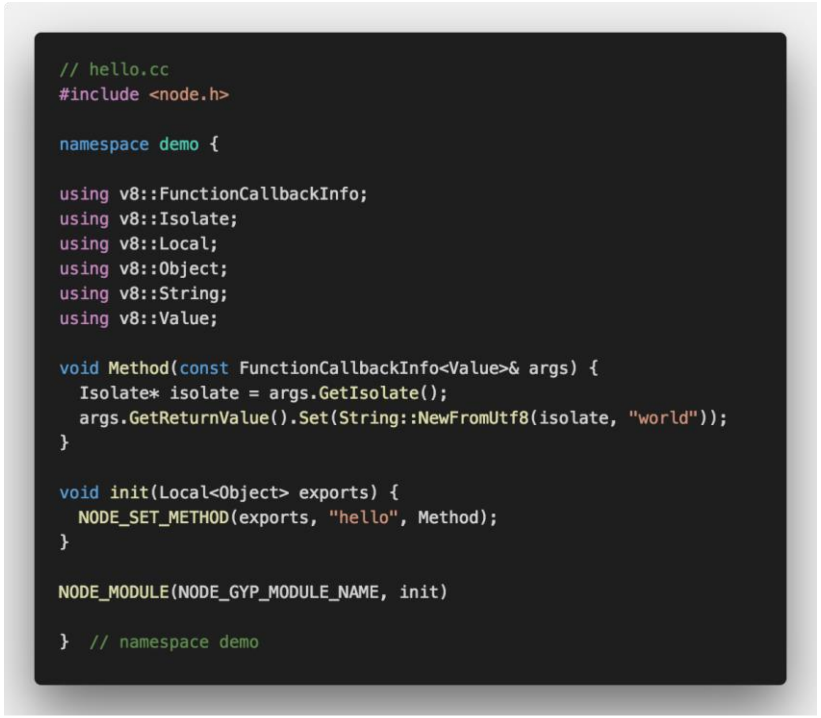
STANDARD MODULES

NodeJS has 2 types of modules:

- C++ Addons
- JavaScript Modules

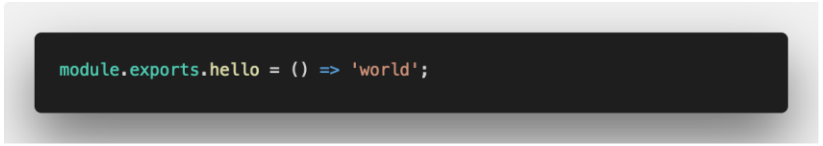
Also you can load JSON files via modules system

STANDARD MODULES. C++ Addons



```cpp
// hello.cc
#include <node.h>

namespace demo {

using v8::FunctionCallbackInfo;
using v8::Isolate;
using v8::Local;
using v8::Object;
using v8::String;
using v8::Value;

void Method(const FunctionCallbackInfo<Value>& args) {
  Isolate* isolate = args.GetIsolate();
  args.GetReturnValue().Set(String::NewFromUtf8(isolate, "world"));
}

void init(Local<Object> exports) {
  NODE_SET_METHOD(exports, "hello", Method);
}

NODE_MODULE(NODE_GYP_MODULE_NAME, init)

}  // namespace demo
```

=



```js
module.exports.hello = () => 'world';
```

https://nodejs.org/api/addons.html#addons_c_addons

STANDARD MODULES. JavaScript Modules

./user.js

```javascript
class User {
  constructor(name) {
    this.name = name;
  }

  greeting() {
    console.log(`Hello, I am ${this.name}`);
  }
}

const developer = new User('Den');
developer.greeting();
```

./index.js

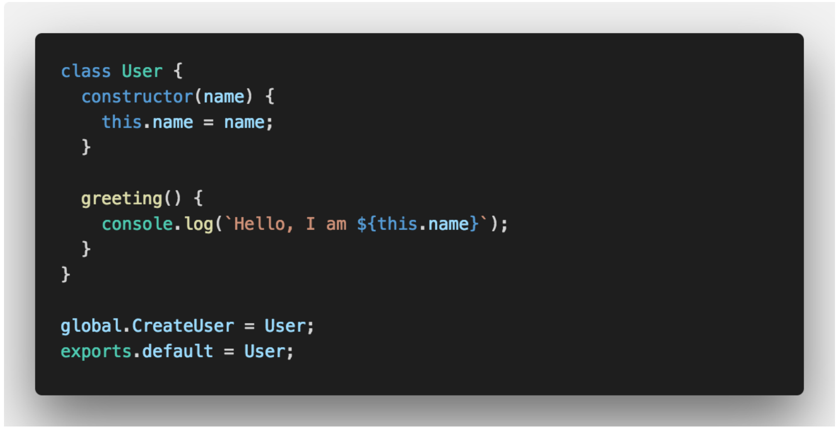
```javascript
require('./user');
```

Result:

```
$ node user.js
Hello, I am Den

$ node index.js
Hello, I am Den
```

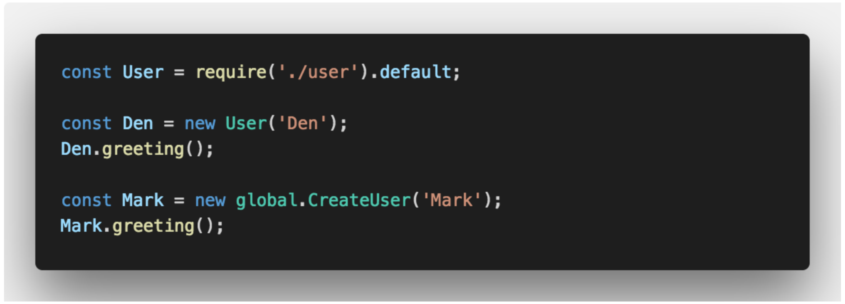
STANDARD MODULES. JavaScript Modules Export

./user.js



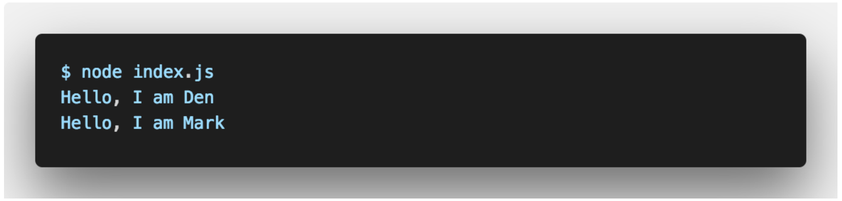
```
class User {
  constructor(name) {
    this.name = name;
  }

  greeting() {
    console.log(`Hello, I am ${this.name}`);
  }
}

global.CreateUser = User;
exports.default = User;
```

./index.js



```
const User = require('./user').default;

const Den = new User('Den');
Den.greeting();

const Mark = new global.CreateUser('Mark');
Mark.greeting();
```

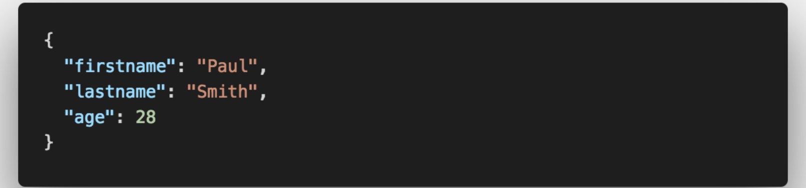
Result:



```
$ node index.js
Hello, I am Den
Hello, I am Mark
```

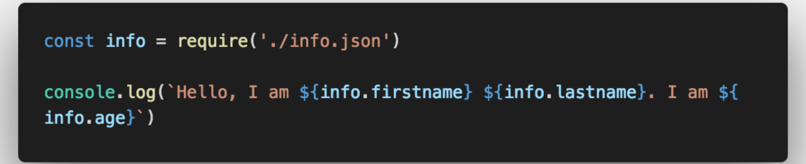
STANDARD MODULES. JSON

./info.json



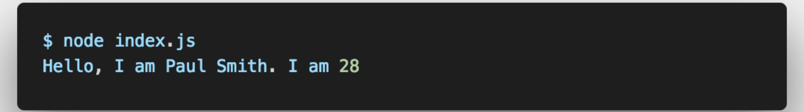
```json
{
  "firstname": "Paul",
  "lastname": "Smith",
  "age": 28
}
```

./index.js



```javascript
const info = require('./info.json')

console.log(`Hello, I am ${info.firstname} ${info.lastname}. I am ${
info.age}`)
```

Result:



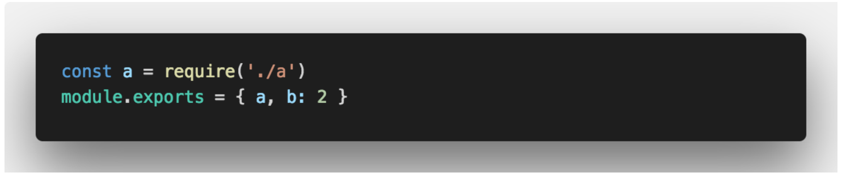
```
$ node index.js
Hello, I am Paul Smith. I am 28
```

MODULE LOAD SYSTEMS

NodeJS has some ways to load modules:

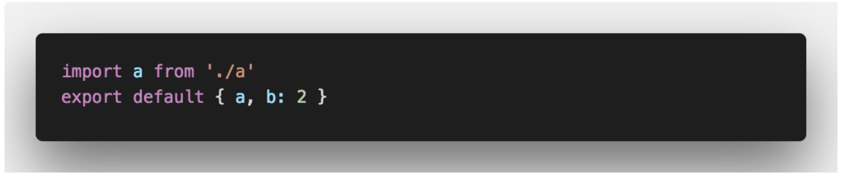
- CSJ (CommonJS)
- ES Modules

CommonJS



```
const a = require('./a')
module.exports = { a, b: 2 }
```

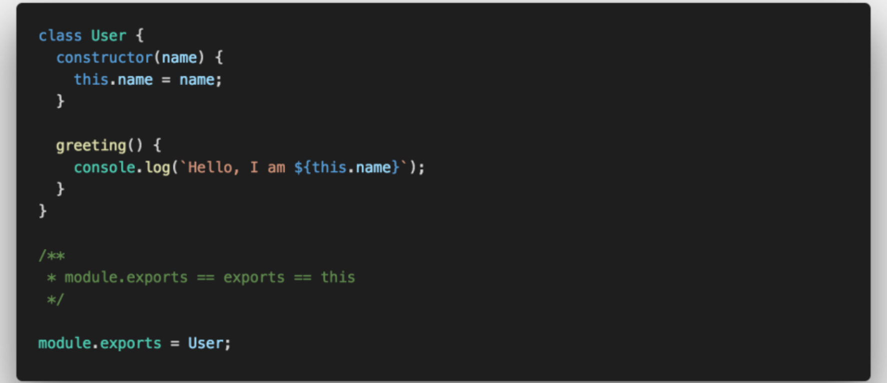
ES Modules



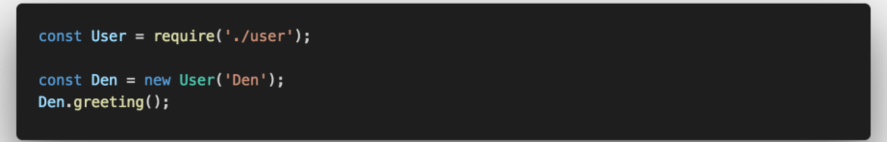
```
import a from './a'
export default { a, b: 2 }
```

DIVE INTO MODULE OBJECT: EXPORTS

CommonJS

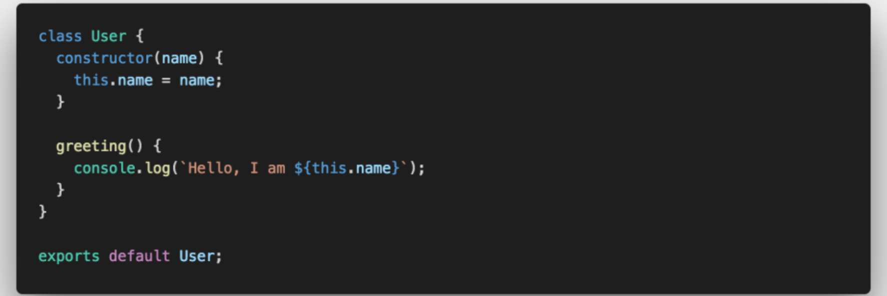


```
class User {
  constructor(name) {
    this.name = name;
  }

  greeting() {
    console.log(`Hello, I am ${this.name}`);
  }
}

/**
 * module.exports == exports == this
 */

module.exports = User;
```

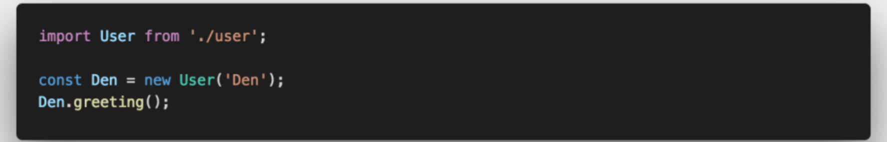


```
const User = require('./user');

const Den = new User('Den');
Den.greeting();
```

ES Modules



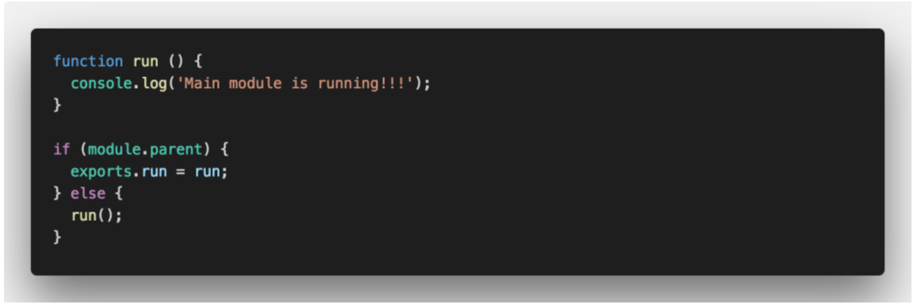
```
class User {
  constructor(name) {
    this.name = name;
  }

  greeting() {
    console.log(`Hello, I am ${this.name}`);
  }
}

exports default User;
```



```
import User from './user';

const Den = new User('Den');
Den.greeting();
```

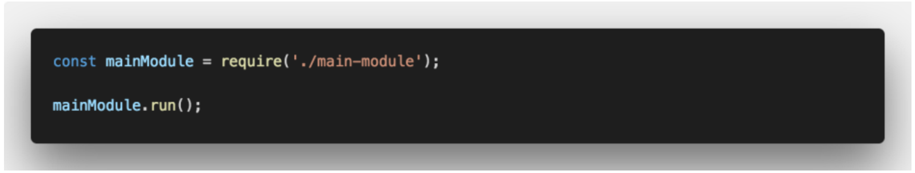
DIVE INTO MODULE OBJECT: MODULE AS APP & COMPONENT

./main-module.js



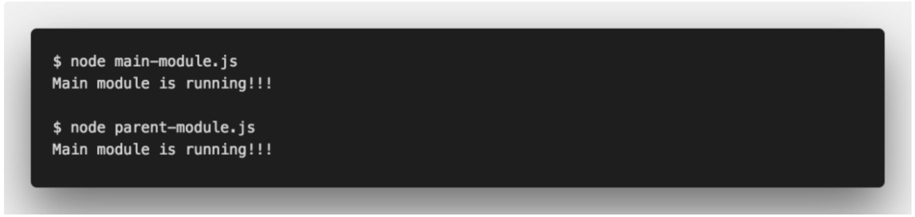
```
function run () {
  console.log('Main module is running!!!');
}

if (module.parent) {
  exports.run = run;
} else {
  run();
}
```

./parent-module.js



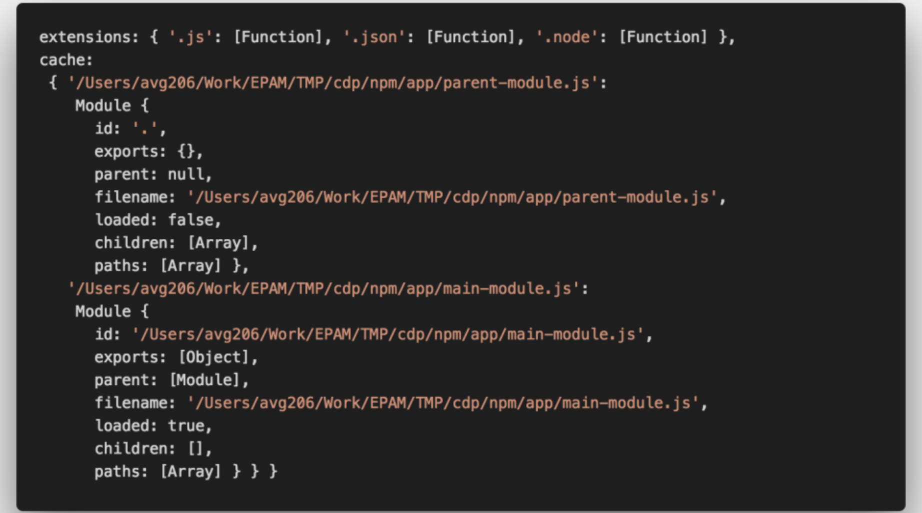
```
const mainModule = require('./main-module');

mainModule.run();
```

Result:

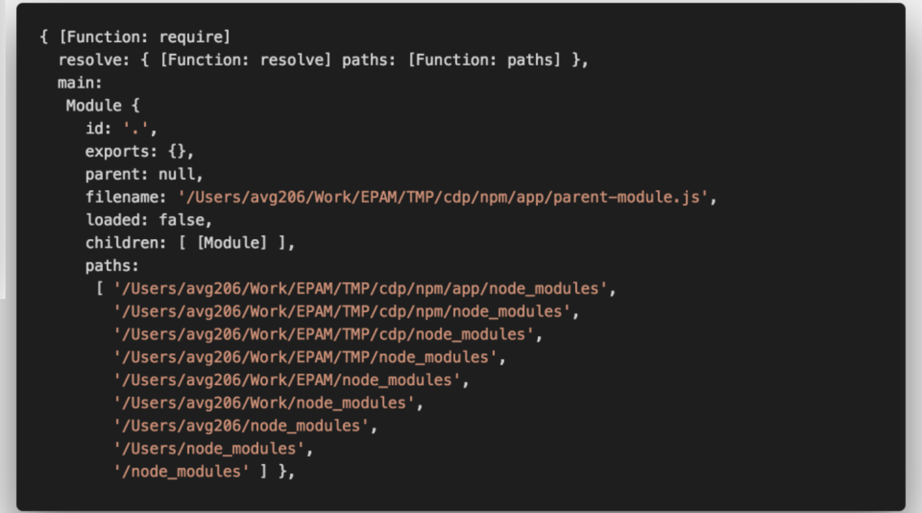


```
$ node main-module.js
Main module is running!!!

$ node parent-module.js
Main module is running!!!
```

DIVE INTO MODULE OBJECT

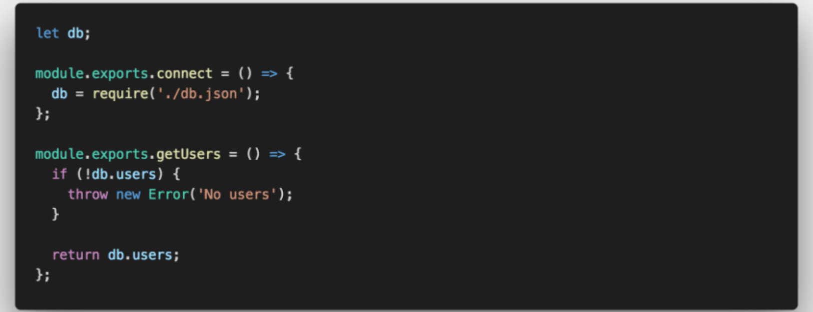


```
extensions: { '.js': [Function], '.json': [Function], '.node': [Function] },
cache:
 { '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/parent-module.js':
    Module {
      id: '.',
      exports: {},
      parent: null,
      filename: '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/parent-module.js',
      loaded: false,
      children: [Array],
      paths: [Array] },
   '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/main-module.js':
    Module {
      id: '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/main-module.js',
      exports: [Object],
      parent: [Module],
      filename: '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/main-module.js',
      loaded: true,
      children: [],
      paths: [Array] } } }
```

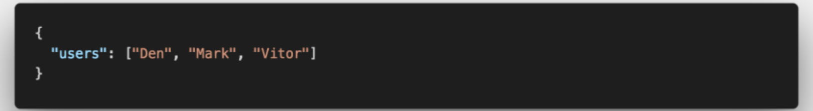


```
{ [Function: require]
  resolve: { [Function: resolve] paths: [Function: paths] },
  main:
   Module {
     id: '.',
     exports: {},
     parent: null,
     filename: '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/parent-module.js',
     loaded: false,
     children: [ [Module] ],
     paths:
      [ '/Users/avg206/Work/EPAM/TMP/cdp/npm/app/node_modules',
        '/Users/avg206/Work/EPAM/TMP/cdp/npm/node_modules',
        '/Users/avg206/Work/EPAM/TMP/cdp/node_modules',
        '/Users/avg206/Work/EPAM/TMP/node_modules',
        '/Users/avg206/Work/EPAM/node_modules',
        '/Users/avg206/Work/node_modules',
        '/Users/avg206/node_modules',
        '/Users/node_modules',
        '/node_modules' ] },
```

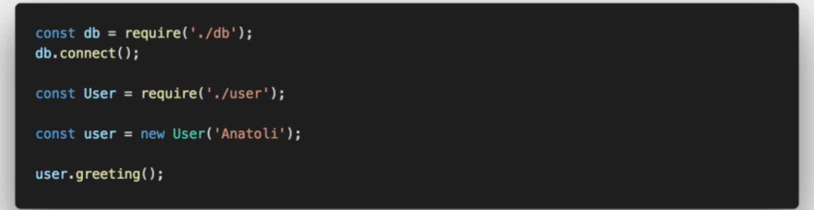
DIVE INTO MODULE OBJECT: CACHING



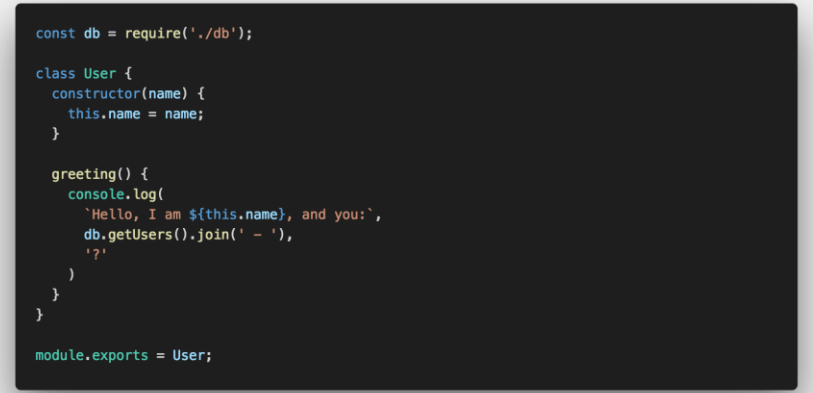
```js
let db;

module.exports.connect = () => {
  db = require('./db.json');
};

module.exports.getUsers = () => {
  if (!db.users) {
    throw new Error('No users');
  }

  return db.users;
};
```



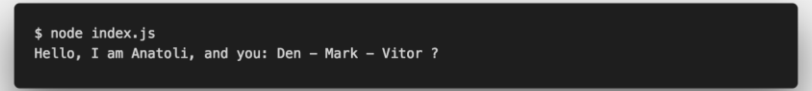
```json
{
  "users": ["Den", "Mark", "Vitor"]
}
```



```js
const db = require('./db');
db.connect();

const User = require('./user');

const user = new User('Anatoli');

user.greeting();
```

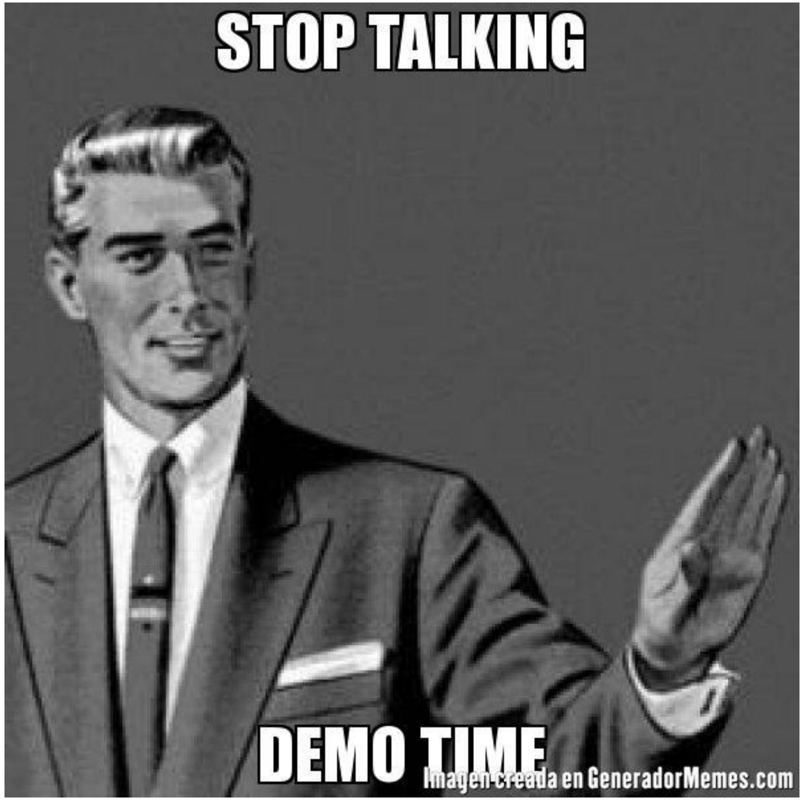


```js
const db = require('./db');

class User {
  constructor(name) {
    this.name = name;
  }

  greeting() {
    console.log(
      `Hello, I am ${this.name}, and you:`,
      db.getUsers().join(' - '),
      '?'
    )
  }
}

module.exports = User;
```



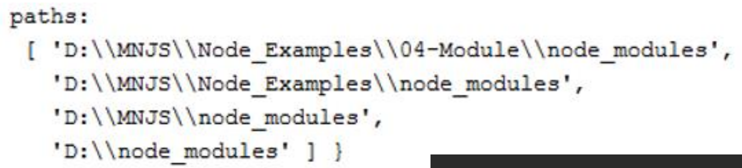
```
$ node index.js
Hello, I am Anatoli, and you: Den - Mark - Vitor ?
```

DIVE INTO MODULE OBJECT: DEMO

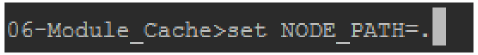


HOW DOES NODE FIND THE MODULE

```
require(X) from module at path Y
1. If X is a core module,
   a. return the core module
   b. STOP
2. If X begins with '/'
   a. set Y to be the filesystem root
3. If X begins with './' or '/' or '../'
   a. LOAD_AS_FILE(Y + X)
   b. LOAD_AS_DIRECTORY(Y + X)
4. LOAD_NODE_MODULES(X, dirname(Y))
5. THROW "not found"
```



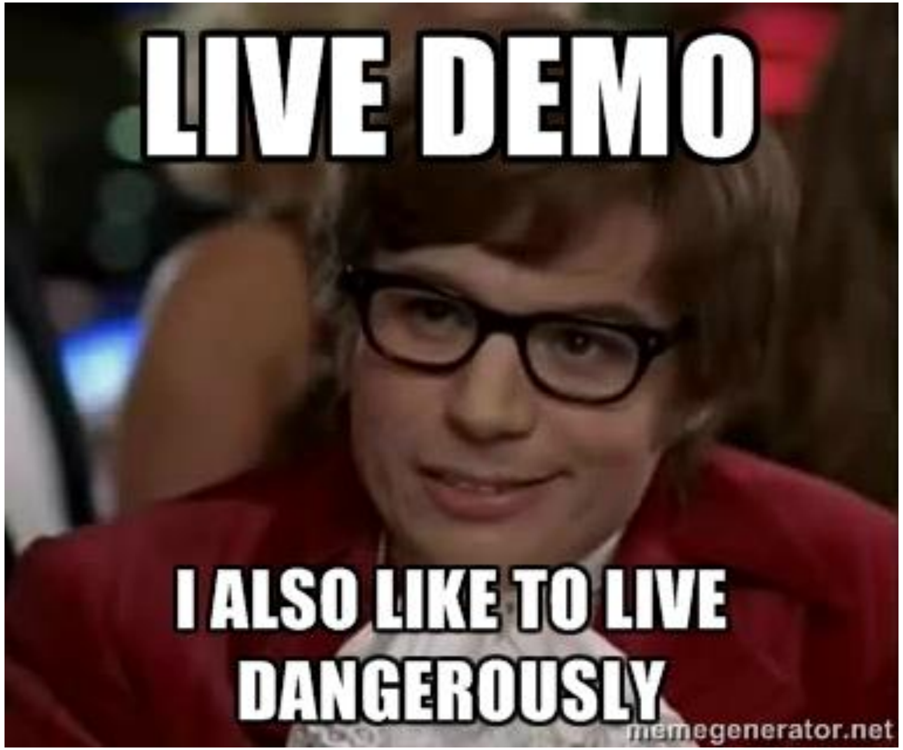
```
paths:
 [ 'D:\\MNJS\\Node_Examples\\04-Module\\node_modules',
   'D:\\MNJS\\Node_Examples\\node_modules',
   'D:\\MNJS\\node_modules',
   'D:\\node_modules' ] }
```



```
06-Module_Cache>set NODE_PATH=.
```

```
LOAD_NODE_MODULES(X, START)
1. let DIRS=NODE_MODULES_PATHS(START)
2. for each DIR in DIRS:
   a. LOAD_AS_FILE(DIR/X)
   b. LOAD_AS_DIRECTORY(DIR/X)

NODE_MODULES_PATHS(START)
1. let PARTS = path split(START)
2. let I = count of PARTS - 1
3. let DIRS = []
4. while I >= 0,
   a. if PARTS[I] = "node_modules" CONTINUE
   b. DIR = path join(PARTS[0 .. I] + "node_modules")
   c. DIRS = DIRS + DIR
   d. let I = I - 1
5. return DIRS
```

HOW DOES NODE FIND THE MODULE: DEMO

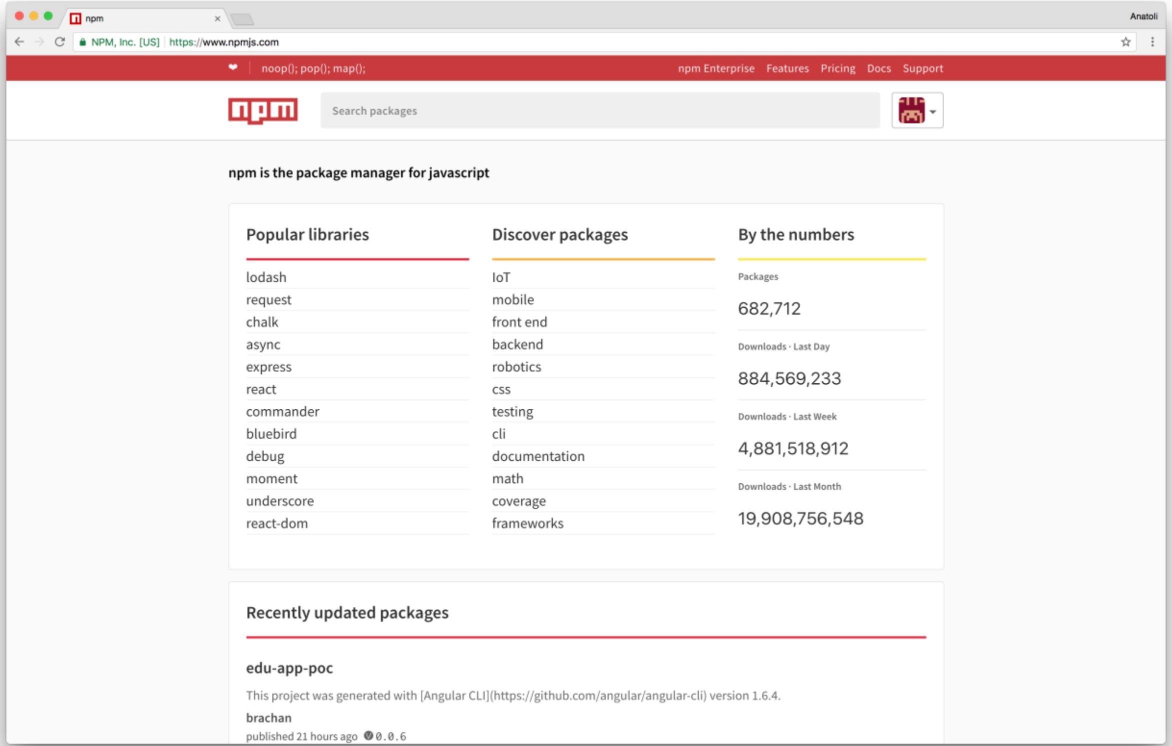


NPM

Agenda (again)

- What is NPM
- Package installation & usage
- Package.json
- NPM CLI
- Create custom package

NPM


npm
Anatoli
NPM, Inc. [US] https://www.npmjs.com
noop(); pop(); map();
npm Enterprise Features Pricing Docs Support
Search packages

npm is the package manager for javascript

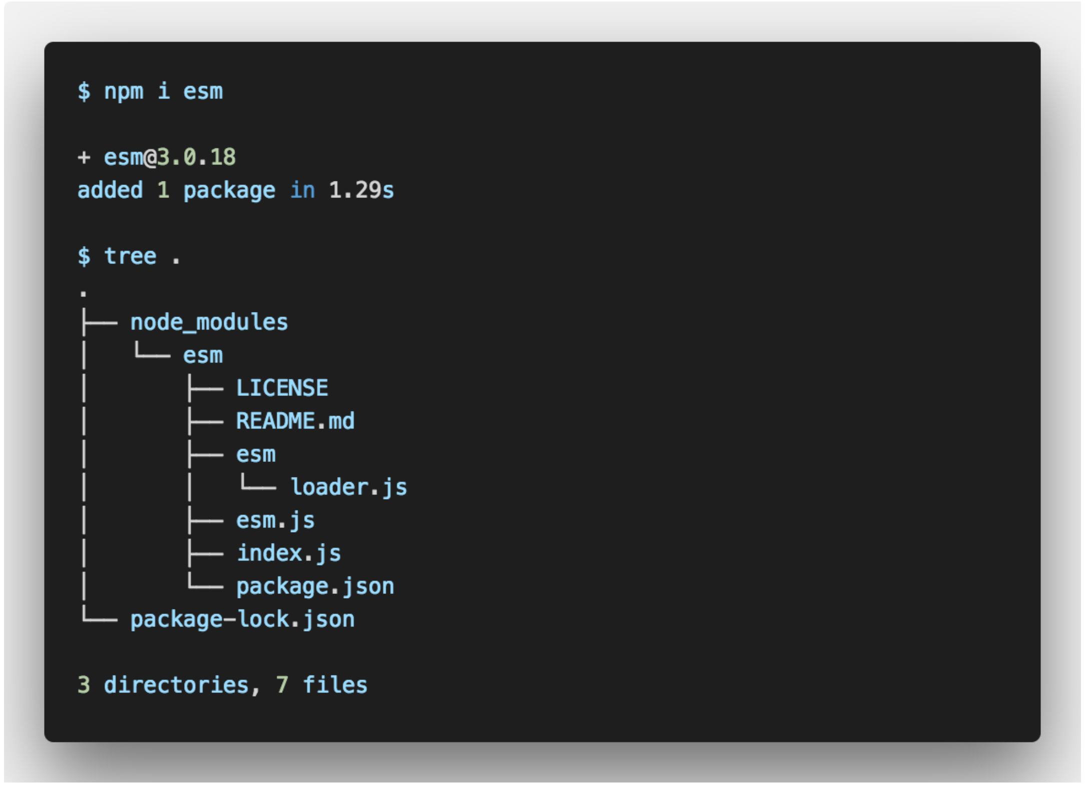
Popular libraries	Discover packages	By the numbers
lodash	IoT	Packages
request	mobile	682,712
chalk	front end	Downloads · Last Day
async	backend	884,569,233
express	robotics	Downloads · Last Week
react	css	4,881,518,912
commander	testing	Downloads · Last Month
bluebird	cli	19,908,756,548
debug	documentation	
moment	math	
underscore	coverage	
react-dom	frameworks	

Recently updated packages

edu-app-poc
This project was generated with [Angular CLI](https://github.com/angular/angular-cli) version 1.6.4.
brachan
published 21 hours ago 0.0.6

NPM: INSTALLATION & USAGE

Simple installation



```
$ npm i esm

+ esm@3.0.18
added 1 package in 1.29s

$ tree .
.
├── node_modules
│   └── esm
│       ├── LICENSE
│       ├── README.md
│       ├── esm
│       │   └── loader.js
│       ├── esm.js
│       ├── index.js
│       └── package.json
└── package-lock.json

3 directories, 7 files
```

NPM: INSTALLATION & USAGE

Why we need *package.json*

```
$ node index.js
internal/modules/cjs/loader.js:550
    throw err;
    ^

Error: Cannot find module 'lodash'
    at Function.Module._resolveFilename (internal/modules/cjs/loader.
js:548:15)
    at Function.Module._load (internal/modules/cjs/loader.js:475:25)
    at Module.require (internal/modules/cjs/loader.js:598:17)
    at require (internal/modules/cjs/helpers.js:11:18)
    at Object.<anonymous> (/Users/avg206/Work/EPAM/TMP/cdp/npm/second/
index.js:1:78)
    at Module._compile (internal/modules/cjs/loader.js:654:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js
:665:10)
    at Module.load (internal/modules/cjs/loader.js:566:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:506:12)
    at Function.Module._load (internal/modules/cjs/loader.js:498:3)
```

./index.js

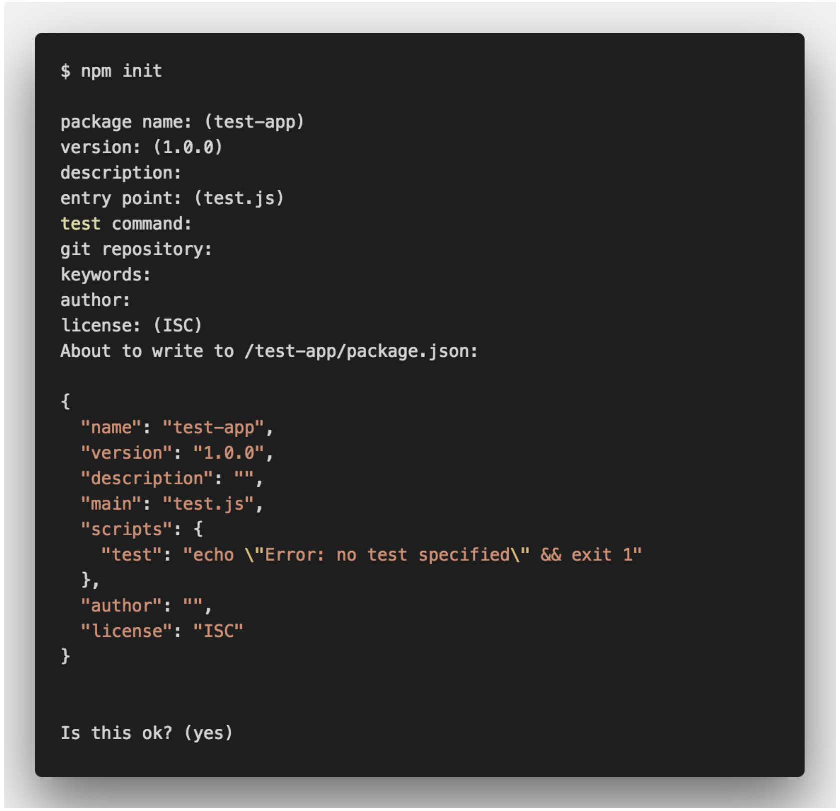
```
const lodash = require('lodash')

const object = { name: 'Den', age: 28, password: 'qwerty' }

console.log(lodash.pick(object, ['name', 'age']))
```

```
$ npm i --save lodash

+ lodash@4.17.5
added 1 package in 4.041s

$ node index.js
{ name: 'Den', age: 28 }
```

NPM PACKAGE.JSON

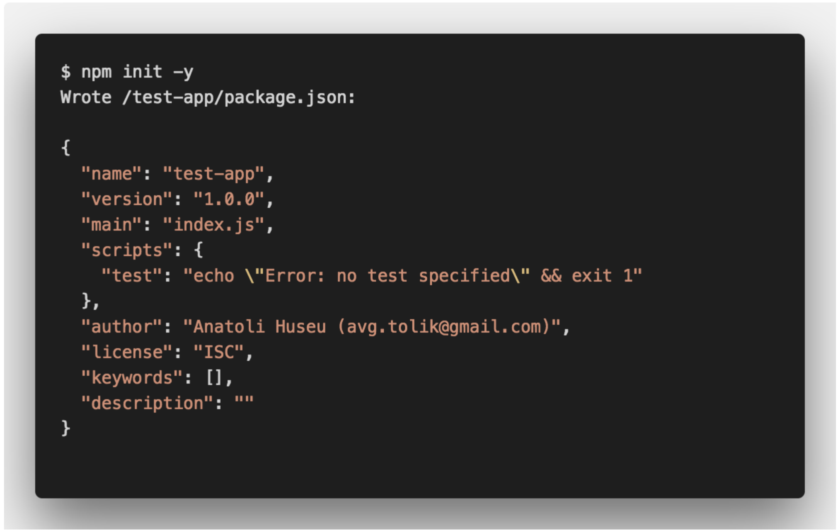
- It serves as documentation for what packages your project depends on.
- It allows you to specify the versions of a package that your project can use using semantic versioning rules.
- Makes your build reproducible which means that its *way* easier to share with other developers.

NPM PACKAGE.JSON: STRUCTURE



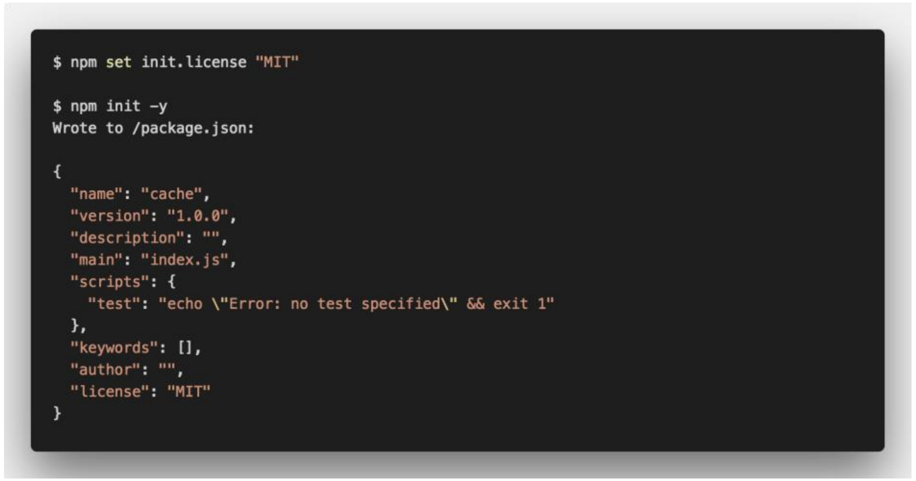
```
$ npm init

package name: (test-app)
version: (1.0.0)
description:
entry point: (test.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /test-app/package.json:

{
  "name": "test-app",
  "version": "1.0.0",
  "description": "",
  "main": "test.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this ok? (yes)
```

OR

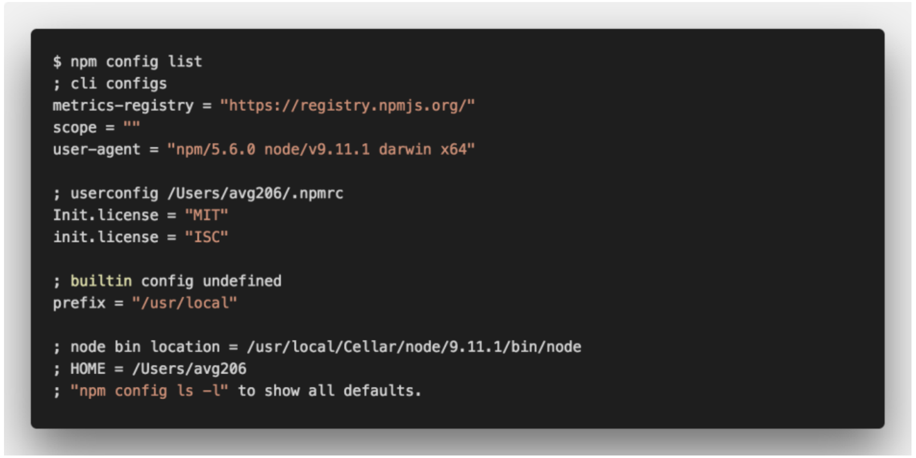


```
$ npm init -y
Wrote /test-app/package.json:

{
  "name": "test-app",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Anatoli Huseu (avg.tolik@gmail.com)",
  "license": "ISC",
  "keywords": [],
  "description": ""
}
```

NPM: PACKAGE.JSON: INIT



```
$ npm set init.license "MIT"

$ npm init -y
Wrote to /package.json:

{
  "name": "cache",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "MIT"
}
```

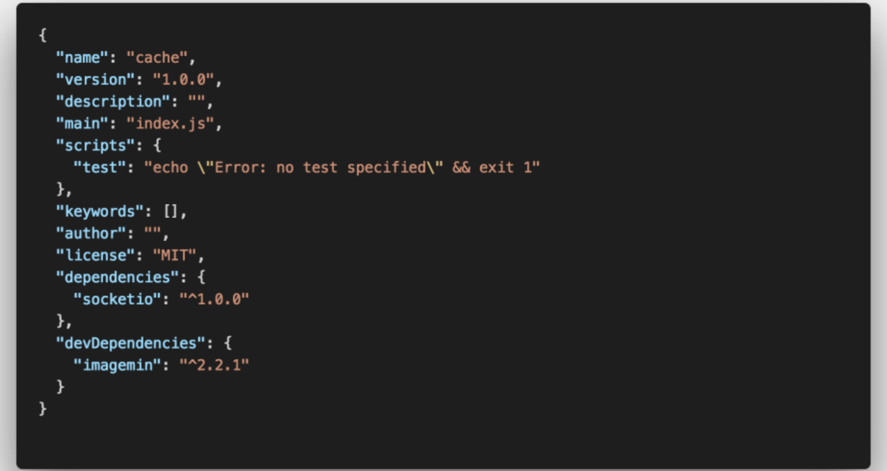


```
$ npm config list
; cli configs
metrics-registry = "https://registry.npmjs.org/"
scope = ""
user-agent = "npm/5.6.0 node/v9.11.1 darwin x64"

; userconfig /Users/avg206/.npmrc
Init.license = "MIT"
init.license = "ISC"

; builtin config undefined
prefix = "/usr/local"

; node bin location = /usr/local/Cellar/node/9.11.1/bin/node
; HOME = /Users/avg206
; "npm config ls -l" to show all defaults.
```

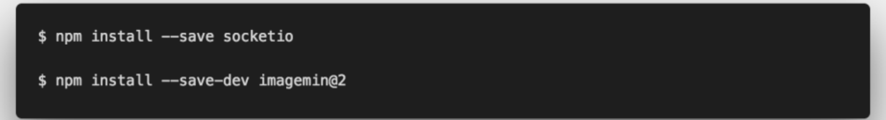
NPM: PACKAGE.JSON: MANAGING DEPENDENCIES

Way to add packages:

1. Edit manually
2. Install with ’--save’, ‘—save-dev’, ‘--save-optional’



```
{
  "name": "cache",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "MIT",
  "dependencies": {
    "socketio": "^1.0.0"
  },
  "devDependencies": {
    "imagemin": "^2.2.1"
  }
}
```



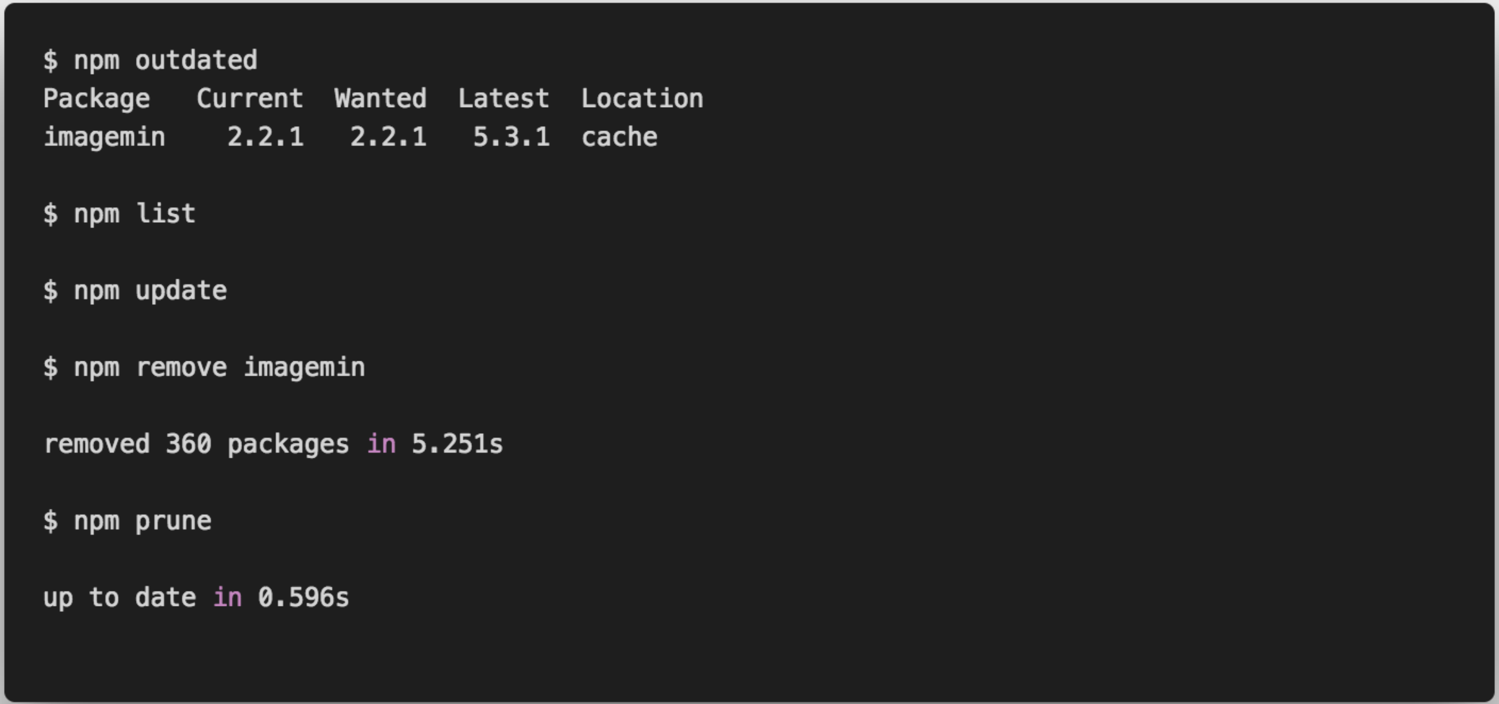
```
$ npm install --save socketio

$ npm install --save-dev imagemin@2
```

NPM PACKAGE.JSON: VERSIONING

Semver (Semantic Version)

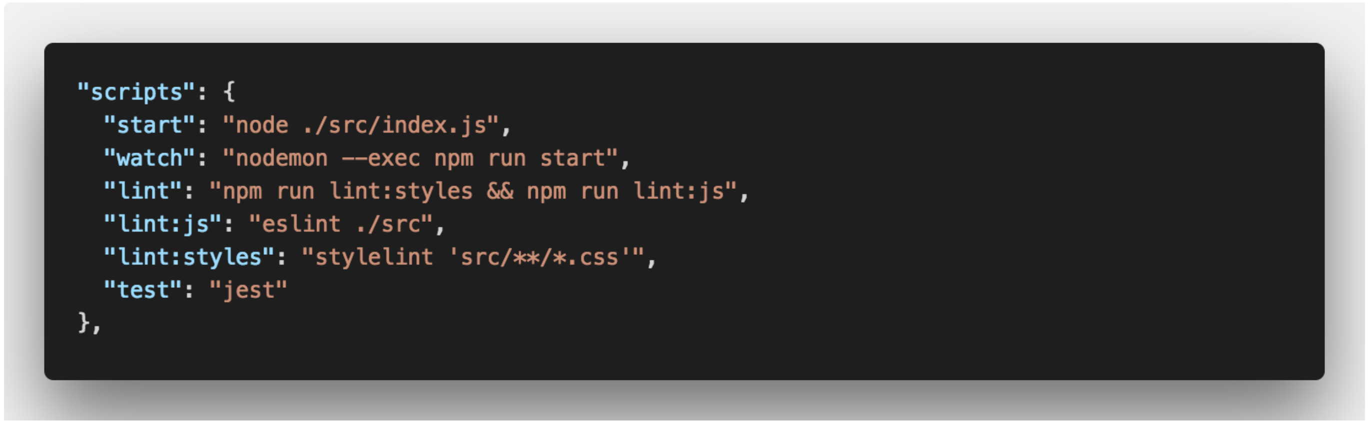
1. Initial release: 1.0.0
2. Bug fixes and other minor changes: Patch release, increment the last number, e.g. 1.0.1
3. New features which don't break existing features: Minor release, increment the middle number, e.g. 1.1.0
4. Changes which break backwards compatibility: Major release, increment the first number, e.g. 2.0.0

NPM: PACKAGE.JSON: MANAGING DEPENDENCIES

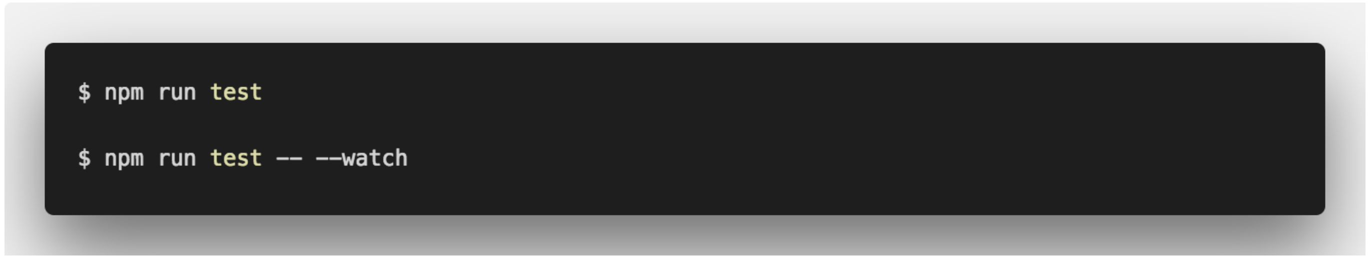


```
$ npm outdated
Package   Current  Wanted  Latest  Location
imagemin    2.2.1   2.2.1   5.3.1  cache

$ npm list

$ npm update

$ npm remove imagemin

removed 360 packages in 5.251s

$ npm prune

up to date in 0.596s
```

NPM: PACKAGE.JSON: SCRIPTS

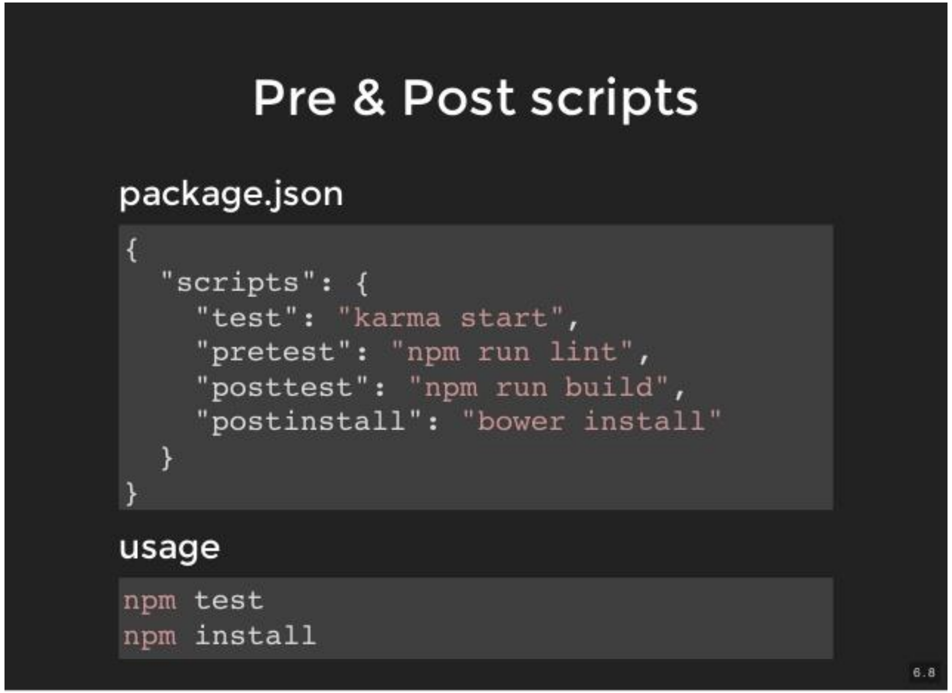


```json
"scripts": {
  "start": "node ./src/index.js",
  "watch": "nodemon --exec npm run start",
  "lint": "npm run lint:styles && npm run lint:js",
  "lint:js": "eslint ./src",
  "lint:styles": "stylelint 'src/**/*.css'",
  "test": "jest"
},
```



```
$ npm run test

$ npm run test -- --watch
```

NPM: PACKAGE.JSON: SCRIPTS



Pre & Post scripts

package.json

```
{
  "scripts": {
    "test": "karma start",
    "pretest": "npm run lint",
    "posttest": "npm run build",
    "postinstall": "bower install"
  }
}
```

usage

```
npm test
npm install
```

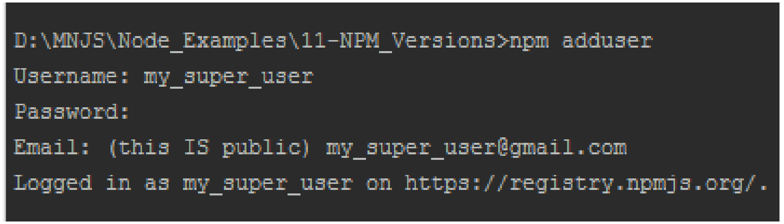
6.8

NPM: NPM CLI

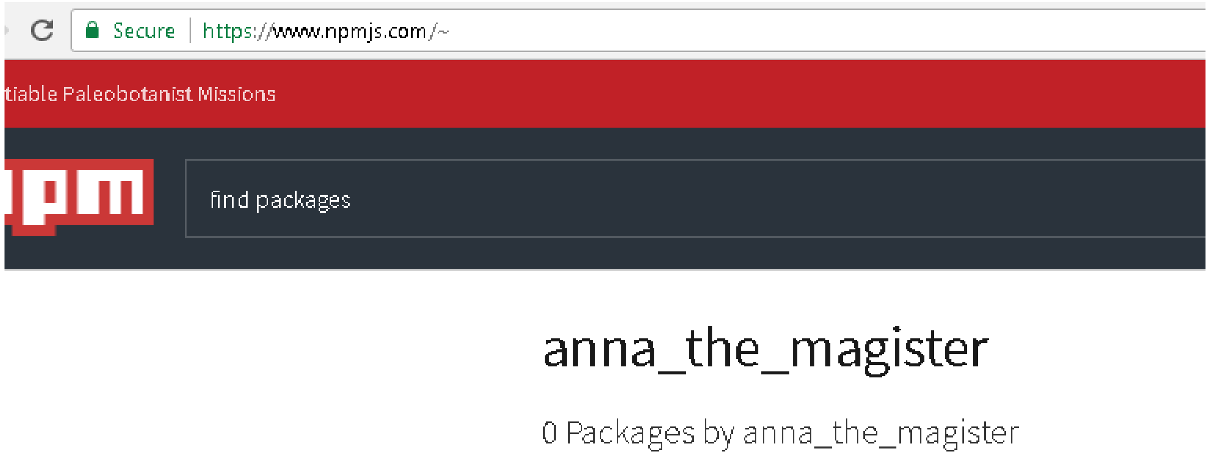
1. npm install [<package>] [--save-dev] [--save]
2. npm init [--yes]
3. npm set <key> <value>
4. npm config list
5. npm list [--depth=<value>]
6. npm outdated
7. npm update
8. npm uninstall <package>
9. npm prune
10. npm help
11. npm dedupe

NPM: CREATE CUSTOM PACKAGE

1. npm adduser / npm login



```
D:\MNJS\Node_Examples\11-NPM_Versions>npm adduser
Username: my_super_user
Password:
Email: (this IS public) my_super_user@gmail.com
Logged in as my_super_user on https://registry.npmjs.org/.
```



Secure | https://www.npmjs.com/~

tiable Paleobotanist Missions

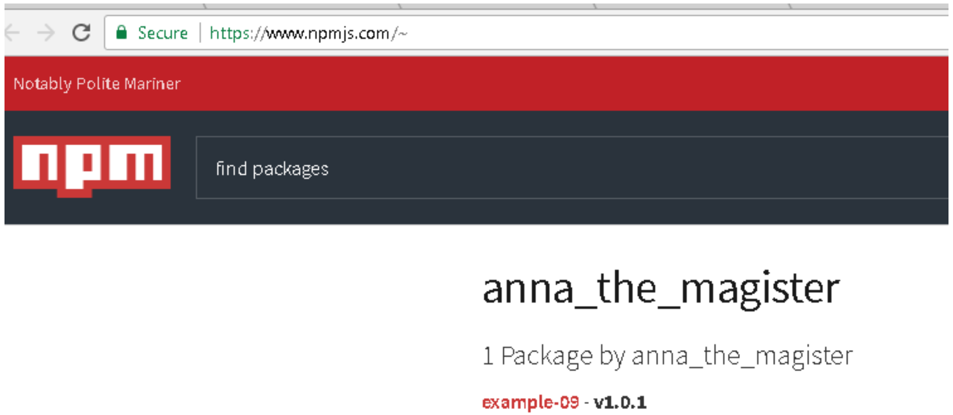
find packages

anna_the_magister

0 Packages by anna_the_magister

NPM: CREATE CUSTOM PACKAGE

2. npm publish

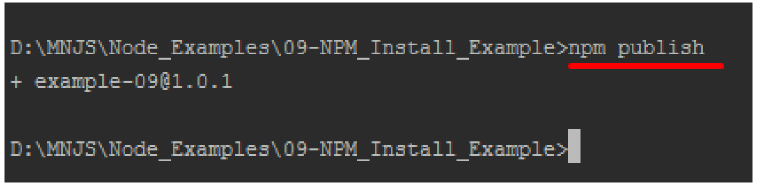

Secure | https://www.npmjs.com/~

Notably Polite Mariner

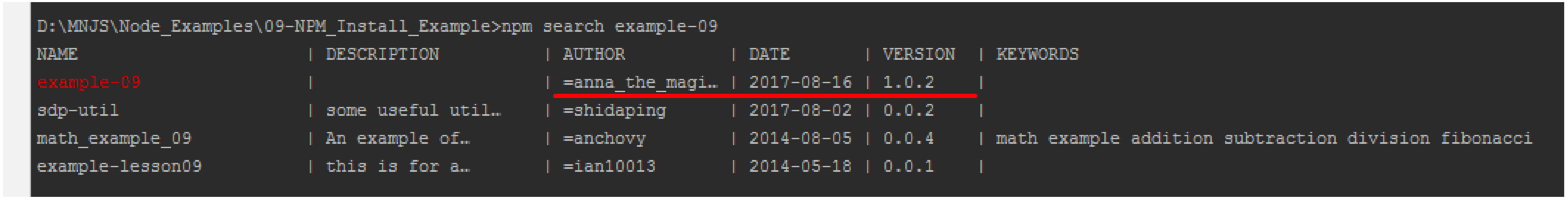
anna_the_magister

1 Package by anna_the_magister

example-09 - **v1.0.1**

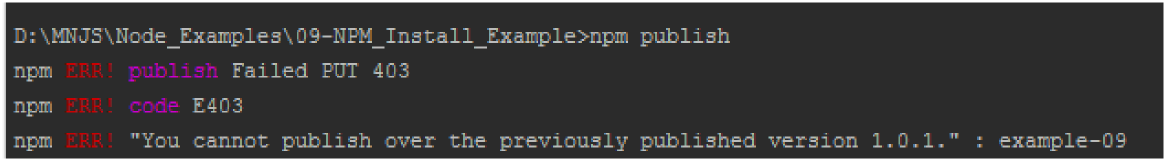

```
D:\MNJS\Node_Examples\09-NPM_Install_Example>npm publish
+ example-09@1.0.1

D:\MNJS\Node_Examples\09-NPM_Install_Example>
```

NOTE: .npmignore & .gitignore

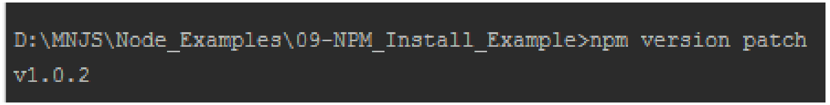

```
D:\MNJS\Node_Examples\09-NPM_Install_Example>npm search example-09
NAME                      | DESCRIPTION          | AUTHOR           | DATE       | VERSION  | KEYWORDS
example-09                |                      | =anna_the_magi… | 2017-08-16 | 1.0.2    |
sdp-util                  | some useful util…   | =shidaping       | 2017-08-02 | 0.0.2    |
math_example_09           | An example of…      | =anchovy         | 2014-08-05 | 0.0.4    | math example addition subtraction division fibonacci
example-lesson09          | this is for a…      | =ian10013        | 2014-05-18 | 0.0.1    |
```

NPM: CREATE CUSTOM PACKAGE

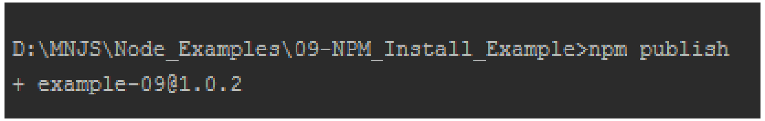
3. npm version patch



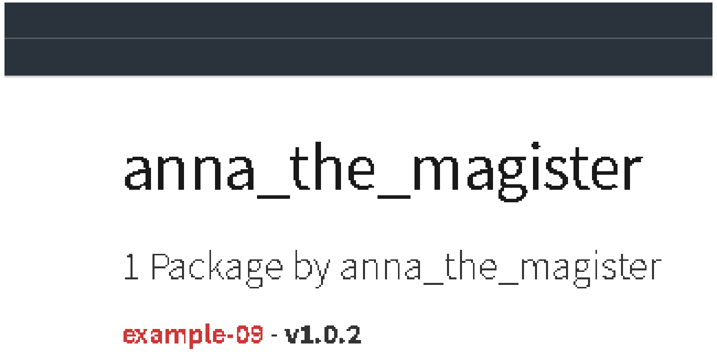
```
D:\MNJS\Node_Examples\09-NPM_Install_Example>npm publish
npm ERR! publish Failed PUT 403
npm ERR! code E403
npm ERR! "You cannot publish over the previously published version 1.0.1." : example-09
```



```
D:\MNJS\Node_Examples\09-NPM_Install_Example>npm version patch
v1.0.2
```



```
D:\MNJS\Node_Examples\09-NPM_Install_Example>npm publish
+ example-09@1.0.2
```



anna_the_magister

1 Package by anna_the_magister

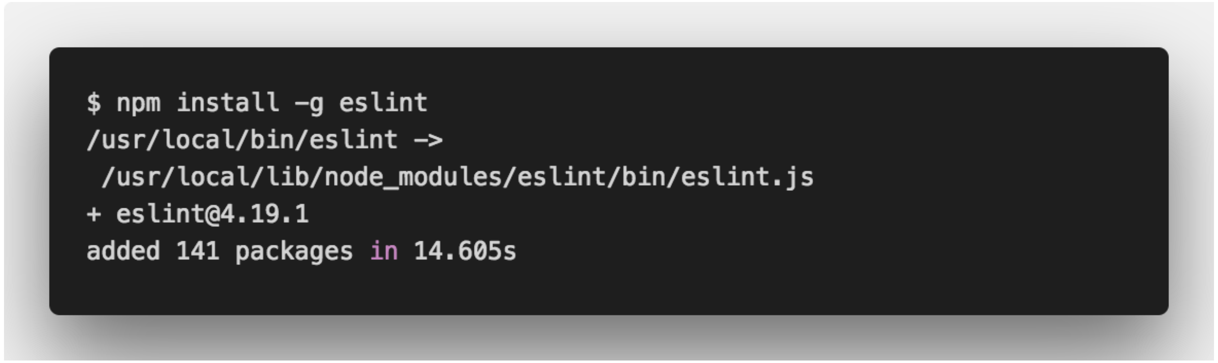
example-09 - v1.0.2

GLOBAL MODULES

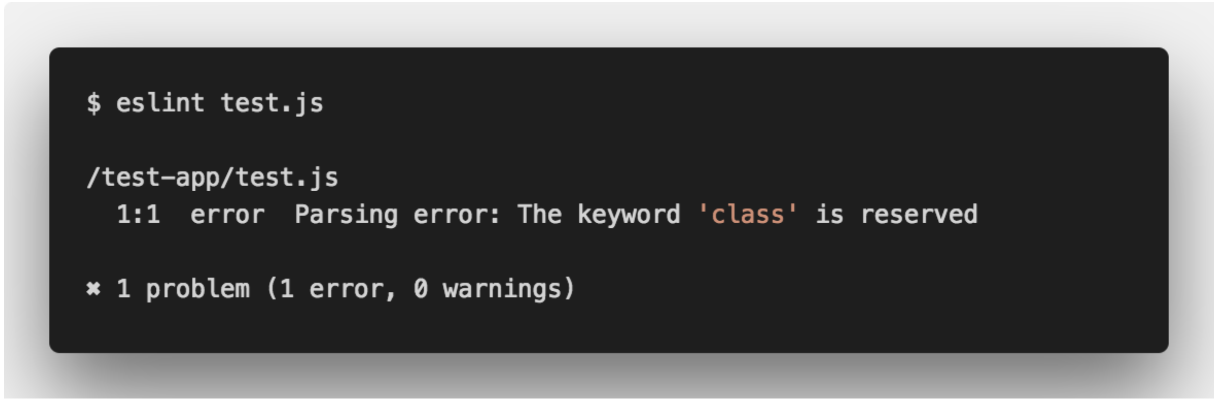
Agenda (again, seriously?)

- Installing packages globally: why and how?
- Global Modules CLI

GLOBAL MODULES



```
$ npm install -g eslint
/usr/local/bin/eslint ->
 /usr/local/lib/node_modules/eslint/bin/eslint.js
+ eslint@4.19.1
added 141 packages in 14.605s
```



```
$ eslint test.js

/test-app/test.js
  1:1  error  Parsing error: The keyword 'class' is reserved

✖ 1 problem (1 error, 0 warnings)
```

GLOBAL MODULES

1. npm update -g <package>
2. npm outdated -g [--depth=0]
3. npm list -g [--depth=1]
4. npm uninstall -g <package>

NPX ??

NPM VERSUS YARN BATTLE

1. Fast installs
2. Offline support
3. Lock file for deterministic installs (untill npm v5)
4. Save to package.json by default

QUESTIONS???

THANKS!

NODEJS MODULES. NPM. YARN.
BY
ANATOLI HUSEU