

NODE- npm

NODE PACKAGE MANAGER

Initialize new project

`npm init`

`npm init --yes`

package.json

```
{  
  "name": "usman-76876567",  
  "version": "1.1.2",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",
```

package.json

```
"dependencies": {  
  "mongoose": "^5.4.0",  
  "underscore": "^1.9.1",  
  "usman-768765674": "^1.1.2"  
},  
"devDependencies": {  
  "jshint": "^2.9.7"  
}  
}
```

npm install underscore

```
var _ = require('underscore');  
const res = _.contains([1, 5, 7, 9], 9);  
console.log(res);  
//node looks in core modules  
//if not found then files or folders  
//lastly in node_modules folder
```

npm uninstall mongoose

Or npm un mongoose

- Will remove mongoose from node_modules folder and will update package.json file also

Source control

git init

git status

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore // add /node_modules

index.js

package-lock.json

package.json

nothing added to commit but untracked files present (use "git add" to track)

Semantic Versioning (Major.Minor.Patch)

Patch: bug fix

Minor: api addition

Major: previous api is modified

"mongoose": "^5.4.0" 5.x

npm package information

`npm list //lists all packages and their dependencies and //versions`

`npm list -depth-0 //lists only packages you installed`

npm package information

npm view mongoose dependencies

npm view mongoose versions

npm i mongoose @2.3.13 //will install a 2.3.13 version

Updating local packages

npm outdated

<u>Package</u>	<u>Current</u>	<u>Wanted</u>	<u>Latest</u>	<u>Location</u>
mongoose	2.4.2	2.9.10	4.13.6	npm-demo
underscore	1.4.0	1.8.3	1.8.3	npm-demo

npm update

//underscore will be updated but not mongoose. Why ?

Updating local packages

```
npm i -g npm-check-updates
```

```
npm-check-updates -u
```

```
npm install
```

Development Dependencies

npm i jshint --save-dev

```
"devDependencies": {  
  "jshint": "^2.9.5"  
}
```

Publishing a package

Make a unique name in package.json. E.g. name="usman1234"

```
>npm adduser //verify your email
```

```
>npm login
```

```
>npm publish
```

Now you can do

```
>npm l usman1234 😊 //your first npm package. Use all exports from index.js
```

Updating a Published Package

`npm publish //` will give error.

`npm version major`

`npm version minor`

`npm version patch`

Babel

```
npm init --yes
```

```
npm i babel-cli babel-core babel-preset-env --save-dev
```

What babel does is

CONVERT THIS

```
// ES6 code using arrow function  
and template literal
```

```
const greet = (name) => `Hello,  
${name}!`;
```

```
console.log(greet('John'));
```

TO THAT

```
// Transformed ES5 code
```

```
'use strict';
```

```
// Equivalent code using a regular  
function and string concatenation
```

```
var greet = function greet(name) {  
    return 'Hello, ' + name + '!';  
};
```

```
console.log(greet('John'));
```

WebPack

Takes CSS/JS/Images and bundle them together

npm i -g webpack-cli

webpack-cli init //answer some questions

npm init

Update package.json add following command

"build": "webpack",