D:\NodeJs\nodeproject-12June>npm init

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields

and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and

save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (nodeproject-12june)

version: (1.0.0)

description: testproject

entry point: (index.js)

test command:

git repository:

keywords:

author: ratnesh

license: (ISC)

About to write to D:\NodeJs\nodeproject-12June\package.json:

{

"name": "nodeproject-12june",

"version": "1.0.0",

"description": "testproject",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "ratnesh",

"license": "ISC"

}

Is this ok? (yes) yes

D:\NodeJs\nodeproject-12June>

First package nodeproject-12June created with two js files:

Index.js

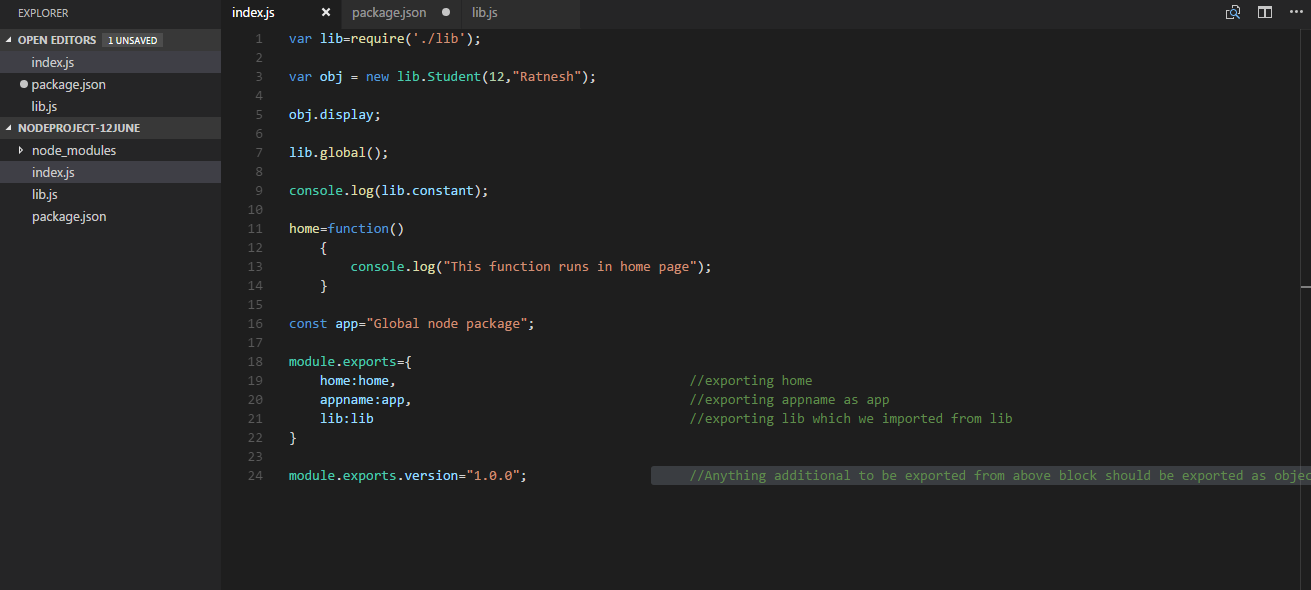
Lib.js

Lib.js

1. Exports three details.

Index.js : imports the information exported by lib. This also exports some details that this added along with exporting lib.js information.

This is then published as a package on website.



var lib= require('./lib');

/\*

var obj=new lib.student(12,"Nick");

obj.display();

lib.global();

console.log(lib.constants);\*/

var home=function(){

console.log("This function runs in the home page");

}

const app="Global node package";

module.exports={

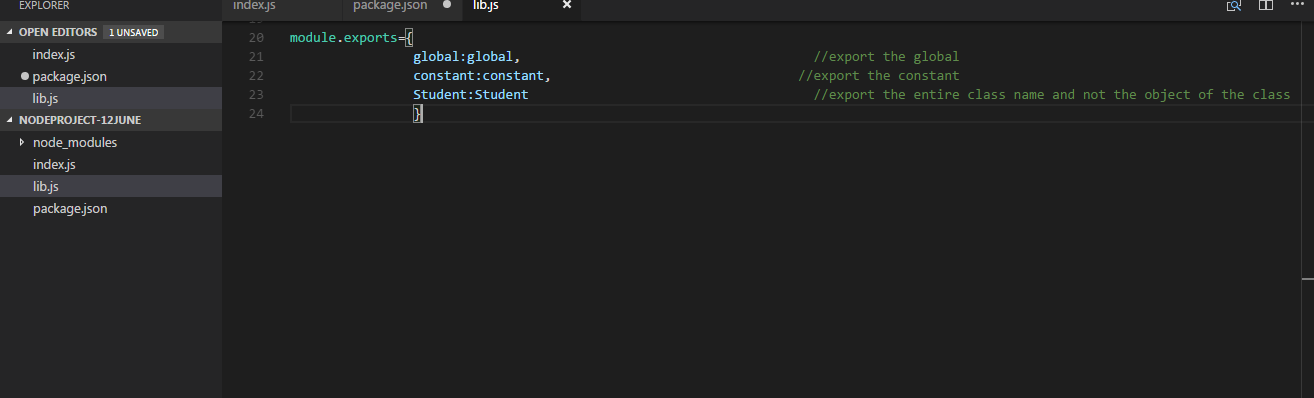
home:home,

appname:app,

lib:lib

}

module.exports.version="1.0.0";



global=function(){

console.log("Global function is running");

}

constants={pf:0.07,hra:0.09,it:0.03};

class Student{

constructor(sno,name){

this.sno=sno;

this.name=name;

}

display(){

console.log(this.sno+" "+this.name);

}

}

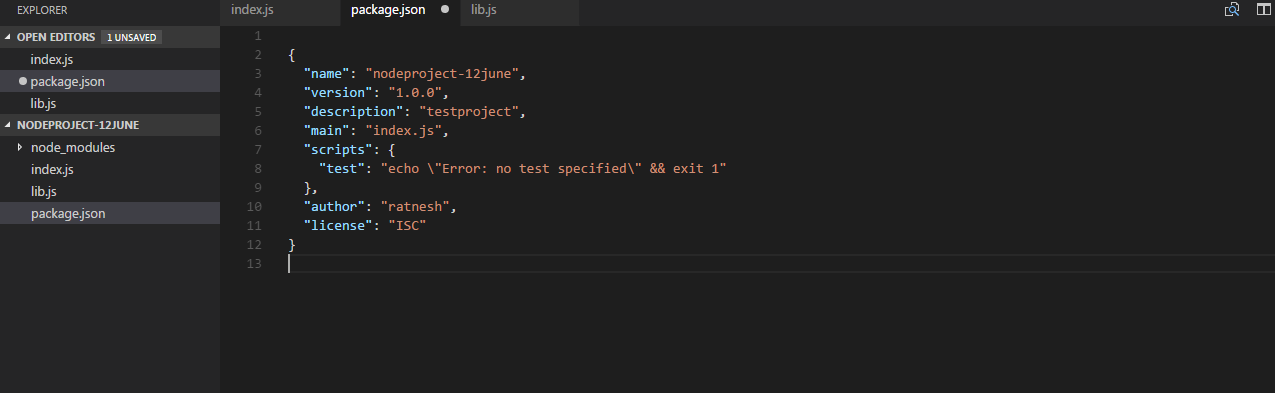
module.exports={

global:global,

constants:constants,

student:Student

}



This can then be published on website by being on the same path of the package.

D:\NodeJs\nodeproject-12June>npm publish ./

npm ERR! publish Failed PUT 403

npm ERR! code E403

npm ERR! you must verify your email before publishing a new package: https://www.npmjs.com/email-edit : nodeproject-12june

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\ratnepan\AppData\Roaming\npm-cache\\_logs\2018-06-12T04\_46\_05\_510Z-debug.log

D:\NodeJs\nodeproject-12June>npm publish ./

+ nodeproject-12june@1.0.0

Create a new package R\_nodeproject-12June\_1:

D:\NodeJs>cd R\_nodeproject-12June\_1

D:\NodeJs\R\_nodeproject-12June\_1>npm init

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields

and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and

save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (r\_nodeproject-12june\_1)

version: (1.0.0)

description:

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to D:\NodeJs\R\_nodeproject-12June\_1\package.json:

{

"name": "r\_nodeproject-12june\_1",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC"

}

Is this ok? (yes)

* Install / download to the local scope the earlier uploaded package in this one so that we can reuse the modules & features being exported from earlier package:

D:\NodeJs\R\_nodeproject-12June\_1>npm install --save nodeproject-12june

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN r\_nodeproject-12june\_1@1.0.0 No description

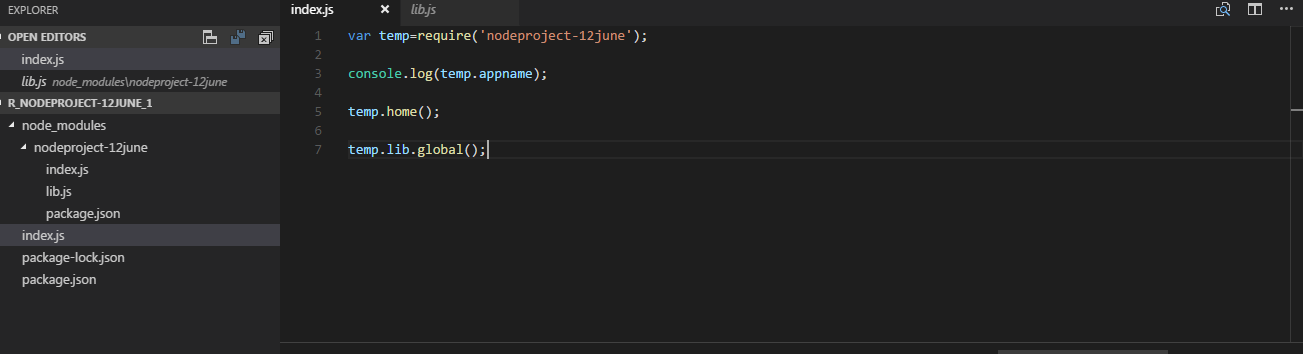
npm WARN r\_nodeproject-12june\_1@1.0.0 No repository field.

+ nodeproject-12june@1.0.0

added 1 package in 3.651s

D:\NodeJs\R\_nodeproject-12June\_1>

* To install a specific version we need to write the command like:
  + npm install –save <name of the package>@<version>



var temp=require('nodeproject-12june');

console.log(temp.appname);

temp.home();

temp.lib.global();

Database Access:

Install Mongodb from the site mongodb.com

We don’t need to install compass.

Environment variable path should have following added if not already present.

C:\Program Files\Git\cmd;C:\Program Files\MongoDB\Server\3.4\bin

mongod is the daemon process for mongodb.

Mongod –dbpath d:\mdata \*\*\*d:\mdata is the path where database is going to be stored.

RDBS (MYSQL) -> ACID -> Atomicity

* consistency.
* isolation

->changes are permanent (durability)

Data:

Structured (DBMS)

Semi structured (

Unstructured. (any other tool probably notepad etc)

SQL database : RDBS and the likes.

noSQL database :

* + columnar database. (Cassandra, Hbase).. Info is stored in column instead of rows.
  + Document based DB (
  + Key value pair database. (redis, membase).
  + Graph based database. (
* MongoDB is a NoSQL Db.
* It is a document based databse.
* Is group of collections (in a rdbms system we call the same as table).
* A collection is group of documents (in a rdbms system a document is equivalent to record).
* Documents don’t follow a fixed schema.
* Any

**Commands**

1. mongo – command can be run when the server is running from a different command prompt.
2. use capgemini – this is to switch to a Db or to create a db.. this also takes the command to this table.
3. show dbs – this shows the databases present.
4. show collections – this shows all the records in a table.
5. db.createCollection(\*\*\*\*\*) ----\*\*\*\* are the values to be entered.

C:\Users\ratnepan>mongo

MongoDB shell version v3.4.7

connecting to: mongodb://127.0.0.1:27017

MongoDB server version: 3.4.7

Welcome to the MongoDB shell.

For interactive help, type "help".

For more comprehensive documentation, see

http://docs.mongodb.org/

Questions? Try the support group

http://groups.google.com/group/mongodb-user

Server has startup warnings:

2018-06-13T14:35:22.643+0530 I CONTROL [initandlisten]

2018-06-13T14:35:22.643+0530 I CONTROL [initandlisten] \*\* WARNING: Access contr

ol is not enabled for the database.

2018-06-13T14:35:22.644+0530 I CONTROL [initandlisten] \*\* Read and wri

te access to data and configuration is unrestricted.

2018-06-13T14:35:22.646+0530 I CONTROL [initandlisten]

2018-06-13T14:35:22.647+0530 I CONTROL [initandlisten] Hotfix KB2731284 or late

r update is not installed, will zero-out data files.

2018-06-13T14:35:22.649+0530 I CONTROL [initandlisten]

> show dbs //only shows admin & local.

admin 0.000GB

local 0.000GB

> use capgemini //created table.

switched to db capgemini

> show collections //nothing present in collections so no output.

> db.createCollection('data') //create a collection with value data.

{ "ok" : 1 }

> show dbs //tried show dbs again and it shows new table

admin 0.000GB

capgemini 0.000GB

local 0.000GB

> show collections // collections now shows collection added above.

data

>

> db.data.insert({sno:1,name:"Ratnesh",city:"Bengaluri"}) //to add record in a collection (table).

WriteResult({ "nInserted" : 1 })

> db.data.find(); //to find records in a collection.

{ "\_id" : ObjectId("5b20e5af0759851c1ef96fea"), "sno" : 1, "name" : "Ratnesh", "

city" : "Bengaluri" }

>

> db.data.find().pretty(); //to see data in more readable format.

{

"\_id" : ObjectId("5b20e5af0759851c1ef96fea"),

"sno" : 1,

"name" : "Ratnesh",

"city" : "Bengaluri"

}

**//importing the data from a json file into DB:**

D:\Users\ratnepan\Downloads\zips>mongoimport --db capgemini --collection zip --f

ile D:\Users\ratnepan\Downloads\zips\zips.json

2018-06-13T15:28:29.080+0530 connected to: localhost

2018-06-13T15:28:29.630+0530 imported 29467 documents

Reading the records from above imported table:

C:\Users\ratnepan>mongo

MongoDB shell version v3.4.7

connecting to: mongodb://127.0.0.1:27017

MongoDB server version: 3.4.7

Server has startup warnings:

2018-06-13T14:35:22.643+0530 I CONTROL [initandlisten]

2018-06-13T14:35:22.643+0530 I CONTROL [initandlisten] \*\* WARNING: Access contr

ol is not enabled for the database.

2018-06-13T14:35:22.644+0530 I CONTROL [initandlisten] \*\* Read and wri

te access to data and configuration is unrestricted.

2018-06-13T14:35:22.646+0530 I CONTROL [initandlisten]

2018-06-13T14:35:22.647+0530 I CONTROL [initandlisten] Hotfix KB2731284 or late

r update is not installed, will zero-out data files.

2018-06-13T14:35:22.649+0530 I CONTROL [initandlisten]

> show dbs

admin 0.000GB

capgemini 0.002GB

local 0.000GB

> use capgemini

switched to db capgemini

> show collections

data

zip

> db.zip.find({city:{$regex:/^Z/}}).pretty();

{

"\_id" : "16063",

"city" : "ZELIENOPLE",

"loc" : [

-80.128564,

40.73136

],

"pop" : 18300,

"state" : "PA"

}

{

"\_id" : "17985",

"city" : "ZION GROVE",

"loc" : [

-76.231032,

40.937152

],

"pop" : 2070,

"state" : "PA"

}

{

"\_id" : "18092",

"city" : "ZIONSVILLE",

"loc" : [

-75.526146,

40.473425

],

"pop" : 2312,

"state" : "PA"

}

{

"\_id" : "19492",

"city" : "ZIEGLERSVILLE",

"loc" : [

-75.485462,

40.281673

],

"pop" : 290,

"state" : "PA"

}

{

"\_id" : "23898",

"city" : "ZUNI",

"loc" : [

-76.810967,

36.843666

],

"pop" : 2162,

"state" : "VA"

}

{

"\_id" : "27597",

"city" : "ZEBULON",

"loc" : [

-78.317368,

35.832078

],

"pop" : 10153,

"state" : "NC"

}

{

"\_id" : "28698",

"city" : "ZIONVILLE",

"loc" : [

-81.747567,

36.319437

],

"pop" : 1633,

"state" : "NC"

}

{

"\_id" : "28790",

"city" : "ZIRCONIA",

"loc" : [

-82.457368,

35.215291

],

"pop" : 2350,

"state" : "NC"

}

{

"\_id" : "30295",

"city" : "ZEBULON",

"loc" : [

-84.310827,

33.100235

],

"pop" : 2470,

"state" : "GA"

}

{

"\_id" : "32798",

"city" : "ZELLWOOD",

"loc" : [

-81.576174,

28.71944

],

"pop" : 1930,

"state" : "FL"

}

{

"\_id" : "33541",

"city" : "ZEPHYRHILLS",

"loc" : [

-82.20566,

28.231063

],

"pop" : 17575,

"state" : "FL"

}

{

"\_id" : "33544",

"city" : "ZEPHYRHILLS",

"loc" : [

-82.34933,

28.263664

],

"pop" : 5989,

"state" : "FL"

}

{

"\_id" : "33540",

"city" : "ZEPHYRHILLS",

"loc" : [

-82.168347,

28.24096

],

"pop" : 15608,

"state" : "FL"

}

{

"\_id" : "33890",

"city" : "ZOLFO SPRINGS",

"loc" : [

-81.742328,

27.480042

],

"pop" : 3515,

"state" : "FL"

}

{

"\_id" : "41396",

"city" : "ZACHARIAH",

"loc" : [

-83.677244,

37.674917

],

"pop" : 36,

"state" : "KY"

}

{

"\_id" : "41397",

"city" : "ZOE",

"loc" : [

-83.668778,

37.666371

],

"pop" : 190,

"state" : "KY"

}

{

"\_id" : "43360",

"city" : "ZANESFIELD",

"loc" : [

-83.664832,

40.302396

],

"pop" : 1171,

"state" : "OH"

}

{

"\_id" : "44656",

"city" : "ZOARVILLE",

"loc" : [

-81.354561,

40.625686

],

"pop" : 2969,

"state" : "OH"

}

{

"\_id" : "46077",

"city" : "ZIONSVILLE",

"loc" : [

-86.276737,

39.956111

],

"pop" : 9759,

"state" : "IN"

}

{

"\_id" : "49464",

"city" : "ZEELAND",

"loc" : [

-86.010408,

42.825586

],

"pop" : 15649,

"state" : "MI"

}

Type "it" for more

>

>

>

> db.zip.find({$or:[{city:"WALES"},{city:"CHESTER"}]})

{ "\_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop"

: 1688, "state" : "MA" }

{ "\_id" : "01081", "city" : "WALES", "loc" : [ -72.204592, 42.062734 ], "pop" :

1732, "state" : "MA" }

{ "\_id" : "03036", "city" : "CHESTER", "loc" : [ -71.244962, 42.967756 ], "pop"

: 2691, "state" : "NH" }

{ "\_id" : "05143", "city" : "CHESTER", "loc" : [ -72.603125, 43.283124 ], "pop"

: 3449, "state" : "VT" }

{ "\_id" : "06412", "city" : "CHESTER", "loc" : [ -72.464293, 41.404903 ], "pop"

: 3417, "state" : "CT" }

{ "\_id" : "07930", "city" : "CHESTER", "loc" : [ -74.677649, 40.789193 ], "pop"

: 6635, "state" : "NJ" }

{ "\_id" : "10918", "city" : "CHESTER", "loc" : [ -74.265116, 41.355381 ], "pop"

: 9196, "state" : "NY" }

{ "\_id" : "19013", "city" : "CHESTER", "loc" : [ -75.374687, 39.849817 ], "pop"

: 49144, "state" : "PA" }

{ "\_id" : "21619", "city" : "CHESTER", "loc" : [ -76.284183, 38.958296 ], "pop"

: 4199, "state" : "MD" }

{ "\_id" : "23831", "city" : "CHESTER", "loc" : [ -77.41557, 37.342934 ], "pop" :

24788, "state" : "VA" }

{ "\_id" : "26034", "city" : "CHESTER", "loc" : [ -80.558439, 40.598109 ], "pop"

: 5966, "state" : "WV" }

{ "\_id" : "29706", "city" : "CHESTER", "loc" : [ -81.21856, 34.714897 ], "pop" :

21920, "state" : "SC" }

{ "\_id" : "31012", "city" : "CHESTER", "loc" : [ -83.174029, 32.398152 ], "pop"

: 1517, "state" : "GA" }

{ "\_id" : "48027", "city" : "WALES", "loc" : [ -82.621728, 43.000058 ], "pop" :

3285, "state" : "MI" }

{ "\_id" : "52134", "city" : "CHESTER", "loc" : [ -92.415528, 43.473021 ], "pop"

: 571, "state" : "IA" }

{ "\_id" : "53183", "city" : "WALES", "loc" : [ -88.378742, 43.008787 ], "pop" :

3171, "state" : "WI" }

{ "\_id" : "57016", "city" : "CHESTER", "loc" : [ -96.975883, 43.898077 ], "pop"

: 799, "state" : "SD" }

{ "\_id" : "58281", "city" : "WALES", "loc" : [ -98.559649, 48.921647 ], "pop" :

122, "state" : "ND" }

{ "\_id" : "59522", "city" : "CHESTER", "loc" : [ -110.97982, 48.454125 ], "pop"

: 1530, "state" : "MT" }

{ "\_id" : "62233", "city" : "CHESTER", "loc" : [ -89.821807, 37.918822 ], "pop"

: 9436, "state" : "IL" }

Type "it" for more

>

> db.zip.find({$and:[{city:{$regex:/^Z/}},{state:"MI"}]}).pretty();

{

"\_id" : "49464",

"city" : "ZEELAND",

"loc" : [

-86.010408,

42.825586

],

"pop" : 15649,

"state" : "MI"

}

>

Updating a record:

> db.zip.find({$and:[{city:{$regex:/^Z/}},{state:"MI"}]}).pretty(); //search an existing row

{

"\_id" : "49464",

"city" : "ZEELAND",

"loc" : [

-86.010408,

42.825586

],

"pop" : 15649,

"state" : "MI"

}

> db.zip.update({\_id:"49464"},{$set:{city:"Zee Land"}}); //run the command for update

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.zip.find({$and:[{city:{$regex:/^Z/}},{state:"MI"}]}).pretty(); //find the row again to see the changes.

{

"\_id" : "49464",

"city" : "Zee Land",

"loc" : [

-86.010408,

42.825586

],

"pop" : 15649,

"state" : "MI"

}

>

To fetch only specific columns:

> db.zip.find({$and:[{city:{$regex:/^Z/}},{state:"MI"}]},{\_id:0,state:1,city:1})

.pretty();

{ "city" : "Zee Land", "state" : "MI" }

Mongoose

HTTP at client side

Get

Post

Delete

put

Server side:

REST service

“Yet Another REST Client” – chrome webstore.