NodeJs

Lightweight

Asynchronous

Cross platform

It can be used in light weight apps. Cannot handle heavy apps.There is only a single thread that listens to the requests. It handles requests Asynchronously.

E book 🡪node.js notes for professionals pdf

tutorialsTeacher.com 🡪node basics

node is a js that can run on server and access server resources

to run a prgm : D:\nodeJS>node hellonode.js

nodemon🡪 helps to monitor nodeapps 🡪CMD 🡪 npm install -g nodemon

Nodemon takes the changes on runtime, we have to run the progm just once.CMD 🡪 Nodemon <filename> any changes made will seen runtime

Npm init cmd makes package.json file with which dependency management will be easy

Node has 3 libs or modules

1.core lib 🡪 will be available during runtime

2.local

3.Third party

Creating a server in node 🡪 we use the core lib ie http

Connecting to DB

Cmd 🡪Npm install MySQL

Mongo DB

No sql data

It has 2 parts: shell and server

Instead of table we have documents

No primary or keys..It has a unique key

Installing mongo🡪 add bin path to envt variables, run mongod command from cmd

1.mongod

2.setting path of the local db : mongod --dbpath "C:\data\db"

3. C:\Program Files\MongoDB\Server\4.0\bin>mongo 🡪run the shell

4. show dbs 🡺 shows all dbs

5. use newdb 🡺 creates new DB

6. show collections 🡺 shows tables in the db

7. db.friemds.insert({name:"priya",location:"ekm",year:"2018"}) 🡺 inserts records & creates the table

8. db.friends.find() 🡺 select \* from friemds

9. db.friemds.find({year:2018}) 🡺 select with where condition

10. db.friemds.find({$and:[{year:"2018"},{location:"ekm"}]}) 🡺 AND contion

11. db.friends.find().limit(3) 🡪top 3 rows

12.db.friends.find().limit(2).skip(1)🡪 skip the first record

13. 12.db.friends.find().limit(2).skip(4)🡪 skip the first 4 recds

14. > db.createCollection('random')

> for(var i=0;i<=100000;i++){

... db.random.insert({digit:i})}

15. db.random.find({digit:99999})

{ "\_id" : ObjectId("5c10f739b3f0f1b7851bebee"), "digit" : 99999 }

>if there are too many records, then finding becomes very slow. In that case we index the records for making search faster

16.Cmd : db.random.ensureIndex({"lottery":1})

It is indexed on lottery.. it can be lottery:1 or lottery:0

Now finding becomes fast

17.sorting based on some key 🡺 db.friemds.find({}).sort({year:1}) 🡺find all the records and sort in ascding order

🡺 db.friemds.find({}).sort({year:-1}) 🡺 desc order

🡺 db.friemds.find({}).sort({year:1}).limit(2).skip(1) 🡺sort 2 recrds skipping first one

18. mongodump --db newdb => create backup of the newdb

Robo 3t tool for db editor 🡪 <https://robomongo.org/>

Documentations : <https://docs.mongodb.com/manual/crud/>