**Phase 3**

**Day 1**

**09-08-2021**

* **Database Management Using MongoDB and Connectivity with NodeJS**

**Node JS**

**Benefits of Node JS**

**Running Node JS program**

**Node JS modules**

**Types of Node JS modules**

**Core module**

**User-defined Module**

**FS Module**

**URL Module**

**Util Module**

**Http Module**

**Creating Web Application http Module**

**Express Third party module**

**Creating REST API using Express Module**

**Get, Post, put and delete methods.**

**Database : No SQL Database**

**Mongo DB Database**

**CRUD Operation Query**

**Mongo DB Relationship**

**Index**

**Aggregate function**

**Connecting Mongo DB database Using Node JS application with the help of**

**Mongodb module and Mongoose Module**

**Creating Express JS with Mongoose module with Mongo DB with Standard MVC.**

**MERN Stack :CRUD Operation**

**Front end -🡪 React JS ----------🡪Express JS -🡪 Mongoose --🡪 Mongo DB**

**Net and socket and web socket programming.**

**HTML/CSS/JavaScript**

**JavaScript library and Framework : using these library or framework we are reading, writing and update DOM properly.**

**JQuery**

**Coffee JS**

**Ext JS**

**React JS**

**Angular JS**

**Angular Framework**

**Vue JS**

**Node JS : Node JS is not a library not a framework. It is a run time environment for JavaScript**

**Library or Framework.**

**Before Node JS we were running JavaScript on browser.**

**Before Node JavaScript is known as Client Side Scripting language.**

**Before Node JS Front end technologies depends upon the back end technologies which develop in other language like Java(Spring boot), Asp.net, Php, Python etc.**

**HTML/CSS/JavaScript/Bootstrap Java**

**jQuery Asp.net**

**Php**

**Python**

**Node JS**

**Frontend backend**

After node JS we can use JavaScript for Client side as well as Server side scripting language.

Using JavaScript code we can do file handling, we can create web application, we can create REST API, we can connect database may be RDBMS (MySQL) and Mongo DB (no SQL Databases), security programming,

Node JS mainly use to do networking and scalable application.

By default JavaScript contains great features ie callack and asynchronous operation.

Non Node JS application if they want to send the data through networking environment.

Like Java, Python, C or C++. Data get block or lock.

Other programming language IO or networking operation data can block or lock.

But Node JS provide non block IO and networking operation with help of callback and asynchronous operation.

Open any terminal

Then write node

It enter in REPL terminal : Read Eval Print Loop

In REPL terminal or Node JS program we can’t use document and window object.

Because document and window object available in Client Side JavaScript programs.

In Node JS DOM and BOM hierarchy.

Server side JavaScript programs we can through command prompt or in REPL terminal.

To exit from REPL terminal cntr +C twice

console is a pre-defined or global object which help to display the output on terminal.

Node JS provided pre-defined object ie process which help to file the processor details about your machine.

Node JS Modules

Node JS modules is a simple or complex functionality organized in a single or multiple JavaScript files. Which can expose using module and we can do re-usability.

Module is like a package in java.

Node JS modules divided into 3 types

1. Core module
2. Local module or user-defined module
3. External module or third party module.

To use any module it may be core, local or external we have to take the help of require function. Which help load the module and we can use the functionality of those module in our application.

fs module (file system module): fs is node js pre-defined core module(by default already available with node JS software). This module help use to do file handling program synchronously as well as asynchronously.

Read and write synchronous operation.

Copy and paste from one file to another file.

Sample.txt targetSample.txt

Node JS

Synch and Asynch

Up to here we store simple data in text format.

We want to store the JSON Data in file .

If we store the data in json it is useful for use to store customer, order, employee, login details.

**Phase 3**

**Day 2**

**10-08-2021**

Fs module with array read, write and append the existing array values.

Node JS provided pre-define d core module is readline. This module is use to read the data from console as asynchronously.

Node js provide third party module ie readline-sync. This module help us to read the value synchronously.

Syntax to install external module

Window user

npm install -g moduleName

sudo npm install –g moduleName

npm install –g readline-sync globally

npm install readline-sync locally

URL Module

url a pre-defined core module. Which provide the client URL details.

<http://localhost:9090/MyWebApp?name=Ravi&age=21>;

<http://www.google.com/index?name=Ravi&age=21>;

in URL reference pare function. if we pass only URL no second parameter the query property consider as a string type.

let urlObj = url.parse(sampleUrl);

If we pass second parameter as Boolean value ie true then query property consider as reference. Which help to retrieve data using queryname.

let urlObj = url.parse(sampleUrl,true);

http module

http is a pre-defined core module which help to create server as well as web application using Java Script programs.

Java : Servlet, JSP and EJB or Spring boot : tomcat, web logic, jboss

Php php is use to create the web application Php we have to run in server : tomcat : apache or Xampp server or any other server.

Asp.net IIS Server

Python Django

Using Node JS with help of http module we can create our own server and run web application.

First we have to load the module using command as

let http = require(“http”);

after loaded the module with the help of reference we have to all createServer function. This function takes callback function as a parameter. So we can pass anonymous function or arrow function.

let server =  http.createServer(function(req,res){

            console.log("Client send the request");

})

The callback function takes two parameter as reference. 1st parameter is consider as request which is use to receive the request from a client and 2nd parameter is consider as response which is use to give the response back to the client.

If we want to make the connection between client to server application it require port number. Which must be free in your machine and it must be unique. Port number is a point where client and server will connect to each other and share the data.

After written the code please run the application using command as

node applicationname.js

Now server is running.

To send the request to server application we have to open the browser

<http://localhost:9090> : URL

res.writeHead(200,{"content-type":"text/html"});

This code is use to write the status code 200 series. It is consider as success.

The content type is html .

Node JS provide lot of pre-defined http request support modules.

https : This module is use to call another REST API Service in node JS application.

request : it is external module which help to call http request which simplified the http or https module.

Request module is not a part of core module so we have to install this module.

npm install request.

First load the module and assign to variable.

Then call function using variableName

request(“url”,{json:true},callback)

callback – err, response and data/body

axios :

when we developing enterprise application

Banking

Loan Section Module

Personal Loan Car/Bike Loan Gold Loan

Java Tech Node JS Python

http module

https/http/request/axios/node-tech

node js can all Java or python or .net REST API.

In Node JS we can use axios module also to call Other technologies REST API.

So first we have to install the axios module.

npm install axios

Difference between Node JS Server Vs other Service like tomcat, IIS, Web Logic etc

Non node Js server is thread base server.

Thread is a small execution of a code with a process.

We will write some task or program using any language may java, python, javascript etc.

After created the program we will give this application to processor to process it.

Processor is responsible to execute the code.

Process : time taken to execute the code or program in execution is known as process.

Process is heavy weighted. Other non node js programming language like Java, Python, .net they provide thread.

Thread is small execution of code with in a process.

Inside process : one or more than one thread can execute.

Multi tasking -🡪 process base

Thread base

Process thread

class Booking {

avl=1;

}

let emp1 = new Booking(); avl =1

let emp2 = new Booking(); avl=1

let emp3 = new Booking(); avl=1

for every client if they create new object

let booking = new Booking();

thread1, thread2, thread3

In Every non js server by default thread base server.

Node JS found limitation in thread base server like tomcat or web logic or jboss.

Tomcat Server : Deploy on online shopping application.

Example sever limit : 1000 thread

1000 client concurrently provide the service.

1001 try to access that time server doesn’t response.

Thread base server can block or lock the request.

Node JS is not a thread base server. Node JS is event loop base server.

Express Module : Express Module is third party module base upon the http module.

Which help to create node JS web application and REST full API very easily.

Express.js- Express for Everyone

Koa.js- Next Generation Node.js Framework

Meteor.js- One Application, One Language

Socket.io- Chat Apps Made Easy with Socket.io

Nest.js- A Nestling of Code

Sails.js- Modernized Data-Oriented MVC Framework

Total.js- A Complete Framework

Hapi.js- Secure than Ever

MERN : Mongo DB , Express React and Node

Express is third party module so whenever we are creating application using express module we have to install this module.

npm install express

in express JS in Get method path must be unique and only one get method can be empty means we can’t write two get method with empty path.

Node Js provide pre-defined global properties ie \_\_dirname. This property provide current directory name.

If we use method =”Get” information send through URL using query params concept.

If we use method = “Post” information send through body part.

In Express js if we want get the data from form post method we have to use another module ie

body-parser. This module provide properties which help to enable to post method data from request.

REST API : Representational State Transfer Application Programming Interface.

Using Express JS we can create the REST API. So those API can call by any technologies like Angular, React JS, Java, Python, Asp.net etc.

So Create Separate Folder Express REST API

npm install express

According REST API we have to use Http Method like

Get

Get means get resource in text format or json format.

Resources means Customer, Employee, Order, Login etc

To retrieve simple message in text format and json format

getInformation in string / test format

if we want to retrieve object then it must be in json format. May be one object or array objects.

getCustomerDetails() json format.

getAllCustomerDetails : array of json .

Client can pass the value to get methods using 2 ways

1. Query param

By default HTML form get method internally use query param concept.

If we want to send only one value then we have to use the syntax as

URL?key=value

If we want to send multiple value then we have to use the syntax as

URL?key=value&key=value&key=value;

1. Path param

url/value1

url/value1/value2/value3

select query

if REST API method use get method

we can call through URL, hyperlink as well as through HTML form with get method.

Post

Post method is use to create the Resource

Store in data in file system or data base system.

Insert query.

Post method we can’t call through URL as well as through hyperlink

We can call post method through form method equal to post.

Browser provide plugin which help to check all REST API call.

Post man client or other plugin

Put

Put method is use to update the resources.

It is equal to update query in database.

Delete

Delete method is use to delete the resource

It is equal to delete query in database

In delete we will delete resource base upon some property cid, empid, accno,srno,.So we will pass this id using pathParams.

Customer, Employee, Product etc.

empId,name,salary, city,job,position etc.

Mongo DB Database

MySQL or oracle or db2 or postgres or firebase

Database system :

Data : raw facts

Information : processed data or meaningful data.

Database : it is use to store the data in table format.(columns and rows).

DBMS : Database Management system : it is a software which is use to store the data in table format using row and column.

RDBMS : Relational database management system.

12 rules DR EF codd’s rules.

All RDBMS database is known schema base database.

Table we have to create and we have defined number of columns and type of value that columns can hold.

Employee

Id Name Salary age phonenumber

100 Raj 12000 null null

101 Raju 14000 null null

102 Ajay 16000 24 null

103 Ramesh null null phone

JSON :

Backend technologies : Java, express, python, php

Data receive from front end technologies in the form json we have to convert into table format.

All RDBMS database use SQL query language.

No SQL database they store data in datable non table formats.

No SQL

Key-value -🡪 redis

Graph database 🡪 Neo4j

Document oriented -🡪 Mongo DB

Column family 🡪 Cassandra and HBase

Mongo DB is type of Document Oriented database. It is open source database. We can store data in document in the form of json.

RDMBS Mongo DB

Database database

Table collections

Records documents.

In table format in json format.

After installation

Please create data folder inside C drive and inside data create db folder.

C:\Program Files\MongoDB\Server\5.0\bin

To run the sever please run the command as

mongod : This command is use to run the mongo db server

mongo : This command open the terminal which help to interact with server.

to check the database from mongo db database.

**show databases : This command is use to display all database**

**Or**

**show dbs :**

use databaseName : This command is use to switch the database. If database is not present it will create and switch to new database else it switch to existing database.

In mongo DB Database is a group of collections like a table.

Syntax to create the collection

db.createCollection(“Sample”);

Mongo Db is a case sensitive database.

To check all collection names from database.

show collections

Or

show tables

In mongo DB we store document in collection like records in tables.

db.collectionName.insert({key:value});

db.Sample.insert({name:”Ravi”})

db.Sample.insert({name:”Raj”,age:21});

you can provide key in double quote or without quote also possible.

Value may be number, string, boolean, array type, complex object, array of complex object.

To view the document from a collection we have to run the command as

db.CollectionName.find();

db.Sample.find();

whenever we store the document in collection mongo db internally created pre-defined properties as \_id which maintains the unique ness between two document.

\_id itself is primary key consider in Mongo DB.

Emp

\_id Name age city

In mongo DB we can insert the record without creating collection.

If collection present it insert the document in existing collection else it create and insert.

Retrieve the document from collection using index position.

db.CollectionName.find()[index];

db.Emp.find()[0];

Get the particular properties value using index position

db.CollectionName.find()[index].property

db.Emp.find()[4].city;

retrieve more than one properties from a documents.

db.CollectionName.find({},{propertyName:1});

1 mean true and 0 means false.

db.Emp.find({},{name:1}); : this command is use to display name properties by default \_id properties also display

db.Emp.find({},{name:1,\_id:0}); : this command display only name property

db.Emp.find({},{name:1,\_id:0,age:1}); : this command is use to display name and age.

db.Emp.find({},{name:1,\_id:0,age:1})[2]; This command is use to retrieve two properties

using index position.

Limit() and skip()

Limit() : this function help to limit to collection to display the number of documents.

db.Emp.find().limit(2);

Skip() : this function help to skip number of document from collection while displaying.

db.Emp.find().skip(2);

sort() : This function is use to display the collection using particular field sort.

1 means ascending and -1 means descending order.

Ascending order by age

db.Emp.find().sort({age:1});

Descending order

db.Emp.find().sort({age:-1});

retrieve the document using conditions like where clause in RDMBS

db.CollectionName.find({condition});

db.Emp.find({\_id:100});

db.Emp.find({name:'Raju'});

db.Emp.find({age:{$gt:25}});

$gt

$lt

$gte

$lte

$eq

$ne

Two more than one conditions.

$and $or operators

$and both condition must be satisfies.

$or any one condition must be satisfies.

db.Emp.find({$and:[{\_id:100},{name:"Raj"}]});

db.Emp.find({$or:[{\_id:101},{name:"Raj"}]});

db.Emp.find({$and:[{\_id:101},{age:{$gt:20}}]});

update documents.

db.CollectionName.update({condition}:{$set:{property:value}});

db.Emp.update({\_id:100},{$set:{age:25}});

db.Emp.update({\_id:100},{$set:{name:'Raj Deep'}});

if condition satisfies more than one document also update query update only one document.

db.Emp.update({city:'Bangalore'},{$set:{age:30}});

if we want to update multiple document we have to use third query braces with multi true option.

db.Emp.update({city:'Bangalore'},{$set:{age:35}},{multi:true});

or

db.Emp.updateMany({city:'Delhi'},{$set:{age:45}});

remove document from a collection

db.CollectionName.remvove({condition})

db.Emp.remove({\_id:14});

db.Emp.remove({name:'Mahesh'})

**Collection Relationship**

**RDBMS**

**One to One : Person -🡪 Passport**

**One to many : Trainer --🡪 Students**

**Many to One : Employees -🡪 Project**

**Many to Many : Students -🡪 Technologies**

**To make the relationship in RDBMS we have to use primary key and foreign key.**

**Trainer\_Student Table**

**TId Tname Tech Sid Sname Age**

**100 Raj Java 1 Reeta 21**

**100 Raj Java 2 Meeta 22**

**100 Raj Java 3 Reeta 23**

**Trainer**

**PK**

**TId TName Tech Student**

**100 Raj Java cell**

**101 Ravi Python**

**Student**

**PK FK**

**Sid Sname Age TSId**

**1 Reeta 21 100**

**2 Meeta 22 100**

**3 Keeta 23 101**

**4 Teeta 24 101**

**In mongo DB we can achieve relationship using two ways**

1. **Embedded style relationship**
2. **Linking style relationship**

**Embedded Style (only one Collection )**

**Employee**

**Id name salary age**

**Address**

**City state**

**One to one relationship**

**{\_id:100,name:”Raj”,salary:24000,age:21,”address”:{city:”Bangalore”,state:”Kar”}}**

**One to many relationship**

**{\_id:102,name:”Ajay”,salary:45000,age:28,”address”:{city:”Bangalore”,state:”Kar”},projects:[{pid:100,pname:”React JS”},{pid:101,pname:”Angular”}]}**

db.Employee.insert({\_id:103,name:"Vijay",salary:42000,age:35,address:{city:"Delhi",state:"Delhi"},projects:[{pid:100,tech:"Angular"},{pid:101,tech:"React JS"},{pid:102,tech:"Python"}]});

db.Employee.insert({\_id:102,name:"Ajay",salary:32000,age:28,address:{city:"Bangalore",state:"Kar"},projects:[{pid:100,tech:"Angular"},{pid:101,tech:"React JS"}]});

condition for complex property

db.Employee.find({"address.city":"Bangalore"}).pretty();

17-08-2021

Linking style relationship. Separate collection.

Trainer

db.Trainer.insert({\_id:1,tname:"Raj",tech:"Java"});

db.Trainer.insert({\_id:2,tname:"Ravi",tech:"Python"});

Student

db.Student.insert({\_id:100,sname:"Reeta",age:21,tsid:db.Trainer.find()[0].\_id});

db.Student.insert({\_id:101,sname:"Meeta",age:22,tsid:db.Trainer.find()[0].\_id});

db.Student.insert({\_id:102,sname:"Teeta",age:24,tsid:db.Trainer.find()[1].\_id});

db.Student.insert({\_id:103,sname:"Leeta",age:26,tsid:db.Trainer.find()[1].\_id});

db.Student.find() : it display student and trainer id details.

Student1

db.Student1.insert({\_id:100,sname:"Reeta",age:21,tsid:db.Trainer.find()[0]});

db.Student1.insert({\_id:101,sname:"Reeta",age:21,tsid:db.Trainer.find()[0]});

db.Student1.insert({\_id:102,sname:"Teeta",age:24,tsid:db.Trainer.find()[1]});

db.Student1.insert({\_id:103,sname:"Leeta",age:26,tsid:db.Trainer.find()[1]});

db.Student1.find(); : it display student as well as trainer document details.

Aggregate functions

Aggregate function is use to combine multiple document from one collection or more than collection. In aggregate function we can use aggregate operator to perform some mathematical operation and return single result.

It is like a group by and having clause.

db.Trainer.aggregate([{$lookup:{from:"Student",localField:"\_id",foreignField:"tsid",as:"StudentsDetails"}}]).pretty();

db.Student.aggregate([{$lookup:{from:"Trainer",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}}]);

group by we have to apply for those properties where value are duplicate like deptId, city etc.

this query makes the group by city

db.Employees.aggregate([{$group:{\_id:"$city"}}]);

Now we can use mathematical operator like sum, avg, max and min.

db.Employees.aggregate([{$group:{\_id:"$city",totalSalary:{$sum:"$salarly"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",totalSalary:{$sum:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",totalSalary:{$sum:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",avgSalary:{$avg:"$salary"}}}]);

db.Employees.aggregate([{$match:{city:"Bangalore"}},{$group:{\_id:"$city",totalSalary:{$sum:"$salary"}}}]);’’’

Connecting Mongo DB database through Node JS application

Node js provide two type of external module is monogdb as well as mongoose module which help to connect the mongo db database.

mongodb is core and native module which help to connect the dabase.

package.json file

package.json is use to maintains all external module dependencies details.

This file also known as configuration file for node js application.

Syntax to create package.json file

npm init

It ask package name : provide any name like com, abc, demo etc

Then enter continuously and give option yes.

Please install all dependencies locally means without –g flag.

If we install without –g package.json file hold the module details using dependencies attribute.

**npm install** : This command is use to download all external module provide in package.json

mongodb module is known as native module which help to do all operation on collection may be insert, delete, update and retrieve documents. Mongodb is good for small application. It doesn’t use schema concept.

Data model (schema). Customer with few pre-defined properties and data types.

mongoose : mongoose is a external module which help to store, retrieve, delete and update from collection as well as it help to developer to create the schema.

Mongoose use for enterprise application.

First load the module

Connect the database

We have to create the Schema :which provide details about the collection.

Base upon schema we have to create the model.

Using model we have to do the operation on documents. Ie insert, delete, update and retrieve.

Create the folder as **node js with mongoose module**

Create package.json file create

Using

npm init

now we have to install

npm install mongoose

Express JS with Mongoose module to connect the Database.

Later we will connect Express JS application through React JS.

Express MVC (Model View Controller).



React JS (View) -----🡪 React JS send the request to backend technologies using axios module with help of get, post, put and delete.

app.js -🡪 Inside this file load all required module like express, bodyparser, mongoose, then connect the database. And run the application in specific port number.

This application check parent path of your request. Then send the request to router file.

router.js : This file check sub path of request and specific methods like get, post, put and delete. This file is responsible to take the decision base upon the sub path and type of methods to do the task.

model.js : This file is responsible to create the schema. Base upon the schema we have to create the model and provide the collection name.

controller.js : Controller will take the help of model file and do the operation on collection base upon the request pass by router file.

Controller file is responsible to do more than one operation on collection with help of model file like insert, delete, update and retrieve etc.

React js(view)---🡪app.js --🡪router.js ---🡪controller--🡪Model --🡪Database(mongoDB)

Collection : to do CRUD Operation

Create new folder MVC

Then create two sub folder backend and frontend

Open the backend folder

Create the package.json file using npm init

**npm install express mongoose**

now create folder

router

controller

model

First we will work in model