Phase 3

Day 1

26-03-2022

Server Side Scripting language ie Node

What is node js

Node js module, core, external module and user-defined module

Fs module

Taking the value through keyboards in Node js

Http module

URL module

Develop the using http module

Express module

Develop the application using express module

Rest Web service using express module

CRUD Operation using Express with Fs module or array module

Calling this application from angular application

Mongo db database

Basic and adv query in mongo db

Connection mongo db using node js with the help of mongo db and mongoose module

Develop application using angular with mongo db with express in MVC style (MEAN Stack)

Socket io programming

Authentication and authorization

Before Node JS JavaScript is known as client side scripting language.

Using JavaScript we were creating front end technologies.

Frontend backend technologies

HTML, HTML5 Java Servlet,jsp or spring boot

CSS, CSS3 and Bootstrap Asp.net

JavaScript and jQuery php

Angular Python

Node JS

JavaScript library and framework

jQuery

backbone js

coffee js

angular js base upon JS 1.x

angular framework base up ts 2 to 13

react js

vue js

D3 js

Node JS : Node Js is not a library or framework it is a run time environment for JavaScript library or framework or application.

Before Node JS we can run the JavaScript program.

After Node JS. Node JS provided lot of pre-defined module with help of those module we can do file handling, creating web application, we can connect database ie mongo db or mysql database using JavaScript.

Before Node JS,JS is known as Client side scripting language. But After node Js JS can be client side a well as server side.

Open the command prompt and hit node

In Node Js program we can’t use document and window object.

Node JS doesn’t provide DOM and BOM

DOM : Document object model

BOM : Browser object model

Node js provided pre-defined global object is console.

REPL : Read Eval Print loop

Node JS Module

Node JS provide module concept. Module contains set of function and properties which help to do specific task. Module is like a container or other language package or namespace.

In node js modules are divided into three types.

1. Core module (by default available with node js software).
2. External module ( we have to install it)

npm install typescript : it is use to enable tsc

npm install –g @angular/cli : ng

1. User-defined module

Fs (File system).

FS is type of core module which provide set of function which help to do file handling program synchronously as well as asynchronously using JS.

If we want to use pre-defined or user-defined module in node js we have to load it

Syntax

let/var refeferenceName = require(“moduleName”);

let fs = require(“fs”);

let obj = require(“fs”);

fs or obj is consider as reference name which help to call pre-defined function available in fs module.

Variable name can be anything.

Core module

fs

net

http

url

path

external module

express : it help to create the web application

connect frontend to backend express using REST API.

mongodb : it is use to connect the mongo db database using node js program

mongoose

cors

Please separate folder fs module – file handling program

Open the vs code in fs module folder

JSON.stringify() : Convert object to string.

The view is normal html page. So this view we can’t do any dynamic task.

To improve view layer Express js provided lot of view engine concept. Those view engine

Provided different syntax to do dynamic task that view page.

Jade

Pug

Etc

Node js provided third party module express-generator which help to create the project with view engine.

npm install express-generator –g

create express generator example folder

To create the new project please run the command as

express demo-app (demo-app) project name

cd demo-app

npm install : install few dependencies present in package.json file.

open the project in vs code

npm start

open the browser this application by default running on port number 3000

<http://localhost:3000>

Now a way day we are not using these all view engine.

Very view engine follow their own syntax.

All view engine tightly coupled with express js application.

Same view engine we can’t use with Java, Python or asp.net

Amazon -🡪 Express Js with any one of the view engine.

Payment ----🡪

Google pay java

XML/JSON Phone pay python

Paytm asp.net

Net banking

May be any other technologies.

Credit card

XML : eXtensible mark up language

DTD, XSD, XPath, XSLT etc

JSON : JavaScript object notation

Web Service : Giving the service for web application when both application running using different technologies.

SOAP base web service : Simple object access protocol. SOAP web service base upon SOA (Service Oriented architecture). Limitation of SOAP base web service we can consume and produce data only in the form xml according the SOAP structure.

Rest full web service : using Rest Full web service we can share the data from one application to another application any format base upon their demand.

We can share xml, json, plain text, html etc.

RESTFull : Representational State Transfer. It is an architecture style to expose our resource as web service so any technologies can communicate with us.

In Node Js using Express module we can create the Rest Full Web Service API (Application programming interface). According RESTfull Web service we can use http protocol methods.

Ie Get, Post, put, patch, delete etc.

Get method : Get resources : Database point of view : get all product/employee/manager/order details.

Get product details by id, by price, etc.

Select query

Post method : create resources : Database point of view : store all product/ employee/ manager / order

Insert query

Put and Patch : update the existing resources :

Put is use to update all properties of existing resources except primary key or unique property

Patch is use to update partial properties of existing resources.

Update query

Delete method : delete resource : Database point of view. Delete the employee/product/manager/customer details using id property.

Delete query

If we make Express JS as Restfull web service then any rest client application can communicate.

It may be

Java, Asp.net, Python, Angular, react JS or JavaScript (fetch or promise) etc.

View ---🡪 Angular Framework

Express Rest Full Web service : here we can create n number of REST API which help to create, delete, update and retrieve all resources.

Sample Rest API Using Express JS

Inside this folder create the package.json file using npm init command

Then install express js

npm install express

Get method

1. Get data in plain text format
2. Get data in json format
3. Data entity(one product details in json format)
4. Data entity (all product details in json format)

If we want to pass the value to REST API

1. Using query param

URL?key=value :if we want to pass one information

URL?key=value&key=value : if we want to pass more than one information

If we use normal html form with method as get internally they use query param concept.

If view is simple html page then you can use query param.

1. Using path param

URL/value1 :if we want to pass one information

URL/value1/value2 : if we want to pass more than one information

If view is rest client application ie angular or react js or any other technologies path param is good.

If want to check post, put, patch and delete method we have use browser plugin

Postman

Arc plugin

Or any rest client plugin

Etc

Day 2

27-03-2022

Taking the value through keyboard using node js

readline : it is type of core module which provide set of function which help to read the value asynchronously.

readline-sync it is a external module which help to take the value synchronously through keyboard.

npm install readline-sync –g

Or

npm install readline-sync

product , id name price

1: Add 2: Delete, 3: Update , 4: display using Id , 5: Display

do {

switch() {

add : pid must be unique

delete using pid

update product price using pid

display using product id

display all

}

}while()

http module : using node js with the help of http module we can create server as well as server side program which help to deal with request and response object.

if we want to create the server side technology like Java, Php , python or asp.net. we require sever.

IIS, Tomcat, Apache etc.

Node JS server is different than other server like IIS, Tomcat, Apache

Difference between Node JS server and Non node js server

Non Node Js server is thread base.

Thread is small execution of a code within a process.

Apache server can take 1000 request concurrently.

Server side using Java, python, C#

class Booking {

avl=1;

}

Booking b1 = new Booking(); avl=1

Booking b2 = new Booking(); avl=1

Booking b3 =new Booking(); avl=1

They create only one memory

Number of client equal to number of thread within same memory.

1st client send request

2nd client send request

3rd client send request

Non node js server like apache, tomcat, IIS. They can block or lock resource when their limit cross.

100 client

If 101 send request to server that sever cant take that request.

Node JS server use event loop concept to handle the request. Node JS is single thread but multi threading.

http module

url : node js provided pre-defined module ie url. Which provide set of function which help to extra the data from url

Day 3

02-4-2022

Using http module we if we create application in node js we have to write more code.

Node JS provide third party web framework modules which help to develop web application using node js.

Express JS is open source web framework provided node js which help to create the web application very easily.

Express JS module internally use http module to create the web application.

First the express js module

In node js project we have to create package.json file. This file hold configuration details about your project.

Syntax to create the package.json file

npm init

it will ask you package name : give some name but don’t give any node js keyword.

Then after that enter the key

Now we have to install the express module

npm install express

node js provided pre-defined global property \_\_dirname. This property provide you current path of folder.

If method is get (by default every form method consider as get). Then data will send through URL using query param concept.

URL?key=value if we are sending one information

URL?key=value&key=value if we are send more than one information.

If method is get. Data is not secure and we can send maximum 255 character data through URL.

If we want to make data secure we have to use method post.

If method post data send through body part of request. Data is secure and we can send huge data to server.

In Express JS we can’t receive value from request body. To receive the value we have to use middleware module (ie intermediate between client and server).

Middleware module name is body-parser. This module by default not available in express js 3.x version.

We were install this module separately using command as npm install body-parser

But from Express JS 4.x version onward this module by default install this express module.

Syntax to use middleware module

app.use(moduleName); app is express module reference which provided pre-defined method is use. Which help to use middleware module depending upon the requirements.

If we want to display alert message in node js application we have to download external module.

alert type of external module which help to display basic alert message in node js.

npm install alert

Day 4

03-04-2022

Express generator

Rest Full Web Service using Express JS

Day 5

09-04-2022

We created REST API

Get : Get resources

1. Get data in string format.
2. Get data in json format
3. Get one product in json format.
4. Get all product details in json format
5. Passing the value for rest api using
   1. Query param

URL?key=value

URL?key=value&key=value

* 1. Path param

URL/value1

URL/value1/value2

1. Find the product details using path param

Post : Store resource

delete method : delete the resource using some property id,name, price

that property we have take through path param.

put or patch method : if we are planning to update all detail using any project then you have to use put . if we are planning do update only few property then we have to use patch method.

patch means partially update

put means complete existing object update.

Angular project to call REST API

Create new project

ng new angular-product-api

routing 🡪 no

styling -🡪 css

npm install –g @angular/cli

or

npm install @angular/cli

please open the angular project in another vs code

then app.component.html open

ng g c product : create the component

ng g s product : create the angular service

ng g interface product : to create the model class or interface

please open model class / interface and write properties which has to map backend objects.

Open the service and do the DI for HttpClient.

HttpClient is a pre-defined API for HttpClientModule. So in app.module.ts file we have to import HttpClientModule in import section.

CORS policy

Cross Origin Resource Sharing

Front end : Angular : This application running on port number 4200 on internal web server provide by angular

Backend : Express Rest API : This application running on port number 3000 using node js server

Two server are going to communicate to each others.

We have to enable this policy in backend technologies.

In node JS to enable this policy we have to use cors module

npm install cors

10-04-2022

Insert the record through angular

Angular forms

1. Template driven form
2. Model driven form or reactive forms

16-04-2022

Storing the data permanently.

1. File base system
2. Database system

Limitation of file base system

1. Data redundancy (duplicate records storing again and again).
2. Data consistency (format of the file).

1,raj,12000

2,ravi,14000

1. Security (read mode or write mode)
2. Doing CRUD Operation (insert, read, update and delete) : difficult.

Database system

Data : raw fact

Information : meaning full data or processed data.

Database : storing the data in table format using row and column.

DBMS : Database management system : it is a software which help to store the data in table format.

Excel sheet can I say simple DBMS.

Employee -🡪 Table

Id Name salary

1 Ravi 12000

2 Ramesh 14000

3 Raju 16000

1 Ravi 12000

Dr. EF Codd’s rules

12 rules : The database which follow 12 rules start from 0 to 11 is known as RDBMS

RDBMS : Relational Database Management System

DBMS like Excel sheet

**TrainerStudentDetails**

TId TName Tech SId SName Age

1 Ravi Java 100 Seeta 21

1 Ravi Java 101 Reeta 22

1 Ravi Java 103 Meeta 23

Trainer

PK

TId TName Tech

1 Ravi Java

2 Ramesh Python

Student

PK FK

SId SName Age TSId

100 Seeta 21 1

101 Meeta 22 1

102 Leeta 23 2

Limitation of RDBMS

All RDBMS database is schema base database.

Table -🡪Employee

Id(int) Name(string/varachar) salary(float) city PhNumber

1 Raj 12000 null null

2 Ravi 14000 null null

3 Ajay 16000 Bangalore null

4 Mahesh 18000 null 991234

No SQL Database

Key-value redis

Graphs database Neo4j

Document oriented mongo db

Column family Cassandra

MonoDb database : Mongo DB is open source document base database which help to store the data in using document with json format.

Before starting mongo db in window OS we have to create data folder in C drive and inside data folder we have to create db folder.

Open the command prompt inside a bin folder of mongo db database.

Open two command prompt

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

In one command prompt run the command as mongod: this command is use to start the

Mongodb database. Wait few second.

Then in another command prompt run the command as

mongo : this command prompt open mongo terminal which help to interact with mongodb database.

We have to write the query in mongo terminal

Ctr + L (this command is use to clear the mongo terminal screen)

show databases : this command is use to display all databases present in mongo db

show dbs

create database databasename this command is valid in mysql but not in mongo db database

use databaseName if database not present it will create and switch to that database else if present it switch to that database.

In mongo Db table is known as collection.

Syntax to create the collection

db.createCollection(“Sample”); This command is use to create the collection.

show collections : this command is use to show the collection

or

show tables : this command is use to show the collection

in mongo db record is known as document.

Syntax to store the document in mongo db database.

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

to view the document from a collection

db.collectionName.find();

in mongo db \_id field is like a primary key. If use or programmer doesn’t pass the value then it internally pre-defined field created for each document ie \_id with unique value.

if you want to pass the value we can pass but we can’t change the \_id field name.

in mongo db we can insert the document without creating the collection.

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

Retrieve specific document using index position

db.Emp.find()[0];

db.Emp.find()[4];

retrieve specific value from a document using index position

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

to retrieve multiple field and with or without condition we can write

db.collectionName.find({condition},{fieldname,fieldName});

retrieve more than one field from a document.

db.Emp.find({},{name:1}); it retrieve name and \_id

db.Emp.find({},{name:-1,\_id:0}); it retrieve only name

db.Emp.find({},{name:1,\_id:0,age:1}); it retrieve name and age

retrieve the document from a collection with conditions.

db.Emp.find({\_id:1})

db.Emp.find({name:"Ravi"});

db.Emp.find({city:"Bangalore"});

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

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

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

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

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

$gt

$gte

$lt

$lte

$eq

$ne

To check both condition

$and

db.Emp.find({$and:[{\_id:1},{name:"Ravi"}]});

db.Emp.find({$and:[{city:"Bangalore"},{age:{$gt:26}}]});

To check any one condition

$or

db.Emp.find({$or:[{city:"Bangalore"},{age:{$gt:26}}]});

sorting

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

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

limit() function it is use to display the top most document from a collection.

Skip() : function is use to display the skip number of document from a collection.

insertMany is use to insert many document at time.

db.Emp.insertMany([{\_id:7,name:"Dinesh",city:"Bangalore",age:34},{\_id:8,name:"Reeta",city:"Delhi",age:29},{\_id:9,name:"Meeta",city:"Mumbia",age:28}]);

update the document from a collection

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

check the condition with \_id ie primary key

db.Emp.update({\_id:2},{$set:{age:25}});

if applying condition with apart from \_id then we have to use updateMany function.

db.Emp.updateMany({city:"Delhi"},{$set:{city:"New Delhi"}})

db.Emp.update({\_id:2},{$set:{name:"Raju"}})

if document with \_id 2 present name field then it will change the field value else it will create new field with new values.

db.Emp.updateMany({},{$set:{desg:"Tester"}});

this query add new field for all document if any document contains desg field then it will change value of that fields.

db.Emp.updateMany({},{$unset:{city:1}});

This query is use to remove all document city fields.

Remove query

db.collectionName.remove({fieldname:value})

17-04-2022

In RDBMS database every cell hold single value.

Student Details

SID SName Age Subject

**SID SName Age Sub1 Sub2 Sub3 Sub4**

1 Ravi 21 Phy Math Che Bio

2 Ramesh 24 Phy Che Bio

**SId SName Age**

1 Ravi 21

2 Ramesh 24

**SubId Subject**

100 Math

101 Phy

102 Che

103 Bio

104 Eng

105 Hindi

**Student\_subject**

SID SubId

1 100

2 100

3 100

Storing array value in collection

db.Subject.insert({\_id:4,name:"Balaji",age:25,subject:["Math","Bio","Phy","Che","Eng","Hindi"]});

rename collection name

db.Subject.renameCollection("Student");

db.Student.find({subject:"Bio"}); : it will search in all index position

db.Student.find({"subject.0":"Phy"}); : it will search in particular index position.

db.Student.update({\_id:1},{$pop:{subject:1}});

Mongo DB relationship

One to one : Employee ---- Address Or Person -🡪 Passport

One to many : Project ---- Employees or Trainer --🡪 Student

Many to one : Students ---🡪 Trainer

Many to many : Employees ---SkillSet or Technologies

One to one : shared primary key.

Person

PK

PID PName

1 Raj

Passport

PK FK

PPID county PI

112233 Ind 1

One to Many : Trainer Student

Trainer

PK

TId TName tech

1 Raj Java

2 Ravi NodeJS

3 Ajay Angular

Select \*from trainer

Student

PK FK

SId SName age TSId

100 Seeta 21 1

101 Meeta 22 1

102 Veeta 23 2

103 Leeta 24 2

104 Teeta 25 null

Select \* from student;

Select t.tname,s.t.tech,s.sname from Trainer t inner join Student s on t.tid = s.tsid

In mongo DB we can achieve same type of relationship using two ways

1. Embedded style relationship : single collection
2. Linking style relationship : more than one collection

Embedded style

First Employee

Employees

Id:1,name:”Ravi”,salary:24000,age:21

Address

City : Bangalore, state : kar

Second Employee

Employees

Id:2,name:”Ramesh”,salary:26000,age:24

Address

City : Bangalore, state : kar

City : Mumbai, state : Mh

Third Employee

Employees

Id:3,name:”Ajay”,salary:28000,age:25

Address

City : Bangalore, state : kar

Project

Pid:1122 tech : Java

Fourth Employee

Employees

Id:4,name:”Ramesh”,salary:32000,age:32

Address

City : Bangalore, state : kar

Project

Pid:1122 tech : Java

Pid:1133 tech : Angular

db.Employee.insert({\_id:1,name:"Ravi",salary:24000,age:21,address:{city:"Bangalore",state:"Kar"}});

db.Employee.insert({\_id:2,name:"Ramesh",salary:26000,age:24,address:[{city:"Bangalore",state:"Kar"},{city:"Mumbai",state:"Mh"}]});

db.Employee.insert({\_id:3,name:"Ajay",salary:28000,age:25,address:{city:"Bangalore",state:"Kar"},project:{pid:1122,tech:"Java"}});

db.Employee.insert({\_id:4,name:"Balaji",salary:32000,age:32,address:{city:"Mumbai",state:"Mh"},project:[{pid:1122,tech:"Java"},{pid:1133,tech:"Angular"}]});

Linking Style

Trainer

\_id (PK)

TId TName Tech

1 Raj Java

2 Ravi Python

Student1

\_id(PK) like FK

SId SName Age Tsid

100 Seeta 21 1

101 Meeta 22 1

102 Leeta 23 2

103 Veeta 24 2

104 Teeta 25 1,2

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

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

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

db.Student1.insert({\_id:101,sname:"Meeta",age:22,tsid:db.Trainer.find()[0].\_id});

db.Student1.insert({\_id:102,sname:"Leeta",age:23,tsid:db.Trainer.find()[1].\_id});

db.Student1.insert({\_id:103,sname:"Veeta",age:24,tsid:db.Trainer.find()[1].\_id});

db.Student1.insert({\_id:104,sname:"Teeta",age:25,tsid:[db.Trainer.find()[0].\_id,db.Trainer.find()[1].\_id]});

In This approach if we want get the Trainer and Student1 details through one collection we have to use $lookup with aggregate function.

Student2

db.Student2.insert({\_id:100,sname:"Seeta",age:21,tsid:db.Trainer.find()[0]});

db.Student2.insert({\_id:101,sname:"Meeta",age:22,tsid:db.Trainer.find()[0]});

db.Student2.insert({\_id:102,sname:"Leeta",age:23,tsid:db.Trainer.find()[1]});

db.Student2.insert({\_id:103,sname:"Veeta",age:24,tsid:db.Trainer.find()[1]});

db.Student2.insert({\_id:104,sname:"Teeta",age:25,tsid:[db.Trainer.find()[0],db.Trainer.find()[1]]});

retrieve the document from more than one collection using aggregate function with $lookup operator.

db.Student1.aggregate([{$lookup:{from:"Trainer",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}}]);

db.Trainer.aggregate([{$lookup:{from:"Student1",localField:"\_id",foreignField:"tsid",as:"StudentDetails"}}]).pretty();

$project : it is use to display specific field with value in aggregate functions.

db.Trainer.aggregate([{$lookup:{from:"Student1",localField:"\_id",foreignField:"tsid",as:"StudentDetails"}},{$project:{tname:1,tech:1,\_id:0,"StudentDetails.sname":1}}]).pretty();

$match : to apply condition in aggregate functions.

db.Trainer.aggregate([{$lookup:{from:"Student1",localField:"\_id",foreignField:"tsid",as:"StudentDetails"}},{$project:{tname:1,tech:1,\_id:0,"StudentDetails.sname":1}},{$match:{tech:"Java"}}]).pretty();

$group : This operator is use to group the documents fields base upon the value and allow to do mathematical operation on those field which contains numerical value. like a group by clause aggregate operator like sum, max, min, avg and count.

db.Employees.insertMany([

{\_id:1,name:"Ravi",salary:24000,deptId:100,city:"Bangalore"},

{\_id:2,name:"Ramesh",salary:22000,deptId:102,city:"Delhi"},

{\_id:3,name:"Rajesh",salary:21000,deptId:101,city:"Bangalore"},

{\_id:4,name:"Ram",salary:20000,deptId:101,city:"Delhi"},

{\_id:5,name:"Ajay",salary:24000,deptId:100,city:"Bangalore"},

{\_id:6,name:"Vijay",salary:28000,deptId:102,city:"Delhi"},

{\_id:7,name:"Mahesh",salary:29000,deptId:101,city:"Bangalore"},

{\_id:8,name:"Lokesh",salary:22000,deptId:101,city:"Delhi"},

{\_id:9,name:"Seeta",salary:25000,deptId:102,city:"Bangalore"},

{\_id:10,name:"Reeta",salary:26000,deptId:102,city:"Delhi"},

{\_id:11,name:"Meeta",salary:27000,deptId:100,city:"Bangalore"},

{\_id:12,name:"Veeta",salary:28000,deptId:101,city:"Bangalore"}

])

Sum of salary for all document.

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

db.Employees.aggregate([{$group:{\_id:"",maxSalary:{$max:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"",minSalary:{$min:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"",avgSalary:{$sum:1}}}]);

db.Employees.aggregate([{$group:{\_id:"",TotalEmployee:{$sum:1}}}]);