**Day 1: 26-11-2022**

**MEAN Stack – Phase 3**

http/htts(req)-----------------🡪

Client Server

🡨---------------------http/htts(res) HTML/HTML5

CSS/CSS3

Bootstrap

JavaScript

jQuery Library

Front end :

DOM : document object model

Jquery is third party library which contains lot of pre defined function which help to read, write and update dom very easily.

Backend technology

Java

JEE (Servlet/JSP/EJB)

Spring framework and spring boot

C# and Asp.net

Python

Php

Node JS

**Node JS** : Node js is a run time environment for JavaScript, JavaScript library or JavaScript framework.

Before Node JS we are calling JavaScript as client side scripting language which help to develop to front end application and we are running JavaScript file using browser with help of HTML. With help of Client side JavaScript we can’t create application which help to store the data in file, storing the data in database (mysql or mongo db), creating rest api etc.

After Node JS we can say JavaScript is also known as client side as well as server side scripting language. So Node JS provided lot of pre defined modules which help to develop server side or backend technology.

<script>

document.write(“Welcome”);

</script>

<script src=”sample.js”></script>

sample.js

document.write(“Welcome”);

With help of node js we can run JavaScript program outside browser means we can run without taking the help of html and browser.

**Difference between library and Framework**

**jQuery is library or React JS**

library is light weighted

library doesn’t follow any standard ie design pattern. Design pattern means best practice.

React JS only follow View.

Library is easy to learn

React js With Redux (to make the standard)

**Angular is Framework**

Framework is heavy weighted.

Framework follow design pattern. MVC architecture or Component base architecture framework.

Model View Controller in Angular Controller is known as Component

View -🡪Component -🡪 Service (Model layer)

Framework is an implementation of design pattern.

Framework is complex.

Angular and React JS help to create SPA (Single page application).

Phase 3 :

Node JS

What is node js why we need to node js

REPL command prompt

Running JavaScript program using node js

Node JS modules

Fs module (file handling module)

Taking the value through keyword using readline module

http module to create web application

express module to create rest api

using with we can create our own service ie get method, post method, put method, delete method , patch method etc.

mongo db database

connecting mongo db database using node js with help of mongodb or mongoose

creating rest api with interacting Mongo db database using MVC style.

module

authentication and authorization

socket io module

Connecting Phase 2 and Phase 3

Angular to Express JS Rest API with FS Module

Angular to Express JS Rest API with Mongo DB

angular and node js connectivity

node –version

npm (node package manager) which help to downloads external node mdoules.

npm install –g @angular/cli

or

npm install @angular/cli --location=global

once we write node command it open REPL terminal

In Node JS program we can’t use document because Node JS doesn’t provide

BOM and DOM hierarchy

Because BOM and DOM is available in browser only not in node js.

In node js in place of document.write we need to use console.log(“Welcome to Node js “);

console is pre defined global object provided by node js

REPL (Read eval print loop)

Client Side JavaScript : no fs module, no http module, no util module, no url module.

Server Side JavaScript: no DOM and No BOM.

But both JavaScript program

Client JavaScript program we can run through html in browser environment

Server JavaScript program we can in REPL or Node command in command prompt.

Node JS provided lot of modules

Module is a collection of more than one function or classes etc.

In Node JS modules are divided into 3 types.

1. Core module (these module by default available with node js software).
2. Third party module : using npm command we need to download those module base upon our requirements.

npm install –g moduleName

1. User defined modules (we can create our own modules).

Node JS help to do networking program in non blocking way.

Because JavaScript program provided great features callback and asynchronous.

FS : file system : fs is core module part of node js which provided set of function which help to do file handling program synchronously as well asynchronously.

In Node js using require() function we need to load the module. If module is core module or built in module we can load easily. If module is external module then first we need to install that module in local machine then we need to use.

var fs = require(“moduleName”);

or

let fs = required(“moduleName”);

Example

let/var fs = require(“fs”);

exception : it is a type of error which generate some think unexpected or abnormal condition occurs during the execution of a program.

try {

one line code more than one line code can generate error

}catch(err) {

console.log(err)

}

Our remaining code execute

**Store or retrieve json data.**

In JavaScript we want to convert Object literal into string format.

JSON.

**Day 2: 27-11-2022**

Node JS provide one of core module ie readline which help to take the value through console or keyboard in asynchronous manner.

Using readline taking the value through keyboard more complex

1. It use asynchronous concept.
2. To take multiple value we need to write nested question functions.
3. Those value are available within that function only.

Node JS provided external module ie readline-sync

This module we need to install. We can install locally or globally.

Which help to read the value synchronously.

npm install modulename locally

after install locally in current path folder node\_module folder download.

Take the value through keyboard using readline-sync and store the data in json file with help of fs module.

**http module :**  hyper text transfer protocol. http is a type of core module which help to create server side web application program using javascript with help of node js. With help of http module we can create our own server and we can run the application which help take the request from client and it will give response back to client.

If we want to create web application using Java (Servlet, jsp or Spring framework), Asp.net, Php, Python etc.

To create the server side technology with help of these language we need to depends upon some server. Because server only provide the benefits to deploy the application which help to take the request and give the response to the client.

Tomcat server

IIS Server

WebLogic

JBoss server

WAS Server

ngInx server

**url** module : uniform resource locator : This is a core module which provide set of function which help to extract details from url.

**Day 3: 03-12-2022**

MEAN

Mongo DB

Express JS

Angular

Node JS

Http is a core module which provide core functionality to develop web application.

Node JS provided lot of pre defined third party module which help to develop web application very easily.

Express JS is a type of external node js web framework module which internally use http module which help to create web application as well as rest api using node js.

Module name is **express**

Express is not a core module which we need to install using npm command.

npm install express –g install globally

Or

Npm install express --location=global

If not working then install locally

npm install express

node js provided \_\_dirname : it is global property which provide current directory path.

From form if we send the data through method as post to express js. Then data will send through request body part. If we want to extract data in express js side we need to enable with help of body-parser external module.

Old version of express js doesn’t contains body-parser module we need install separately. But in New version express js module automatically install all required module to develop the application using express module.

Alert is pre defined function in client side JavaScript part of window object.

In Node JS we can’t use BOM and DOM.

Node js provided external module

alert

so we need to install this module in application to use it

npm install alert

**Web Service**

**REST API**

**REST API Using any entity like Employee, Customer, Order, Product**

**Array or file Storing in File**

**Day 4: 04-12-2022**

Express JS engine

If we use pug or jade are Express js view engine. Those engine help us to do dynamic task on view page.

LoginServer.js

Node JS provided external module ie express-generator which help to create express js project with standard syntax with any one of the view engine.

So first we need to install express generator

**npm install –g express-generator**

or

**npm install express-generator**

express

to create the project using express generator we need to run the command as

express demo-app

cd demo-app move inside a project

then install few dependencies

npm install

plz open this project in vs code

to run this project we need to run as

npm start

by default this project run in default port number 3000

If we use Jade View Engine Then our application must always we need to use View as Jade.

But If we use any Express JS view engine then our application view and backend technology they tightly coupled to each others.

Servlet Express JS

JSP HTML / Jade

Java

Client Amazon Payment

Paytm java

Google pay node js

Phone pay asp.net

Net banking php

Credit card python

Web Service : Giving the service for web application when both the application running using different technologies may in same os or different os.

SOAP Base Web Service Simple Object Access Protocol.

RESTFull Web Service Representational State Transfer

Xml/json

HDFC HSBC

Xml/json

Java Node JS Express

eXtensible Markup language

JavaScript Object Notation

In SOAP web service we can share the data between two technologies only in the form of xml.

RestFull web service : We can consume and produce data in any format base upon client requirement. Like XML, JSON, text, html etc.

But if we make express JS as RESTFull web then view can be any technologies like Java, Asp.net, Php, Angular, React JS or any rest client.

npm init command is use to create package.json file

npm install command read package.json file from this file it read all dependencies details and download in local machine.

Creating Rest API for Entity Example Employee, Customer, Product, Manager, Order, Account etc.

Employee entity or object

Post method : Create the resource like Employee, Order, Customer

In REST API we need to write post method which help to consume the data in the form json and we can store in array or file or database.

Create new resource

If we connect to database then we need write insert query

Get method : Get the resource like Employee, Order, Customer etc

In REST API we need to provide URL which help to get all information about that resource and we need to get the specific resource details base upon any property

Get All Employee, Customer, Product details,

They can pass empid then if present provide employee details else give information employee details not available.

If we connect to database when we need to write select query

We can get all details

We can find details using specific property

1. Using query param : URL/path?key=value single value

URL/path?key=value&key=value multiple value

By default html form with get method internally use query param concept.

1. Using path param

URL/path/value1 single value

URL/path/value1/value2 multiple value

Put method : Put method is use to update the resource. We need to provide rest api which help to update the resource details base upon any property like update name, salary using employee id etc.

Updating existing resource

If we connect to database when we need to write update query

Patch method : Patch method also use to update the existing resource.

Patch is use to update partial resource update and put is use to pull existing update.

Employee Id,Name,Age,Salary : we need create resource using post method

If we are planning to update age or salary or name then we need to patch. We are changing only few property.

If we are planning to update name,age,salary means all property using id property then we need to us put method.

Delete method this method is use to delete the resource using any property like id, accno,srno etc.

We can check get method through Browser URL.

But we can’t check post, put, patch and delete method throws browser.

We need Rest Client plugin to check those methods.

Arch plugin for browser

PostMan rest client plugin

**Day 5: 10-12-2022**

Get method : get all resource : all product details

Get product details using / passing pid

Using query param

Using path param

Post method : store or create the resource

Patch or Put : to update the records

Delete : to delete the records

Storing the data in file system using fs module

Limitation of file system

1. Data redundancy (means we can store same data again and again or duplicate record).
2. Data inconsistency (format of the file). .txt, pdf, doc, json, xml etc.

txt, pdf, doc

id,name,salary

1,ravi,12000

Id name salary

1 ravi 12000

Id/name/salary

1/ravi/12000

excel : it is use to store the data in table format.

json or xml : json has own format, xml has own format.

1. Security of the file (we can provide the security for the file only read mode or write mode or execute file).
2. Doing CRUD Operation Create or insert, read, update and delete are complex.

Database : Using Database we can store the data permanently.

Data : Raw fact.

Information : Processed data or meaningful data.

Database : it is software which help to store the data in table format.

DBMS : Database Management system Excel is small DBMS software. Because excel allow us to store the data in table format.

RDBMS : Relational Database Management system

Excel sheet

Employee

EId EName Salary

1 Ravi 12000

2 Ramesh 14000

2 Ajay 16000

4 Ramesh 1800abc

RDBMS :Relational Database Management System

RDBMS software provided us syntax to create the table structure (schema)

What is the table name, table contains number of columns and with their data types.

Employee ,

Id(number/int), Primary key (if column is primary key that column doesn’t allow duplicate value as well as null (empty value)).

Name(varchar(10)/string),

salary (float/double/decimal) etc

StudentTrainerDetails

Sid SName Age TId TName Tech

1 Seeta 21 100 Raj Node JS

2 Reeta 22 100 Raj Node JS

3 Meeta 23 101 Ravi Java

4 Leeta 24 101 Ravi Java

Student

PK

Int varchar(50) int FK(link to trainer table pk ie tid)

Sid SName Age TSId

100 Seeta 21 1

101 Reeta 22 1

102 Meeta 23 2

103 Keeta 24 null

Trainer

PK

Tid TName Tech

1 Raj Node JS

2 Ravi Angular

If we want retrieve the records from more than one table rdbms software provide us join concept.

MySQL, Oracle, Db2, PostGres, SQL Server 2022 etc.

Foreign key : FK is use to link to primary key of same table or another. If column FK that column allow only those values which present in primary key. It can allow duplicate but doesn’t allow any other value apart from primary key values. It can allow null value.

All RDBMS Database is known as Schema base database. Means before storing the data in database. First we need to create table with name as well as number of columns with their data types.

Employee

PK

Id Name Salary age City

100 Ravi 10000 null null

101 Ramesh 12000 null null

102 Raju 14000 21 null

103 Lokesh 16000 null Bangalore

Retrieve the data from single table is faster than retrieve the data from multiple table using join.

Angular supporting JSON.

Express JS allow use to consume json data from front end technologies.

To convert each json data into table format is more complex.

Can store the data in any other format rather that in table format.

RDMBS is SQL (Structured Query Language) which help to interact with database to store, retrieve, update and delete the records from database.

NoSQL Database came in picture which allow store the data in any other format rather than table format.

Always data categories in three format.

Structured format (table format )

Semi structured format (jons, xml)

Un unstructured format.

No SQL Database

Key-value Redis

Graphs data Neo4J

Document oriented Mongo DB

Column family Casandra

Mongo DB : Mongo DB is an open source document oriented database which help to store the data in document in the format of json. Performance wise mongo db is faster than other RDBMS Database.

RDBMS Mongo DB

Database Database

Database is know as Database is known as

group of tables. group of collections.

table collection

record(tuple) document

column field

In table format in json format.

In Window OS

In C Drive

Create data folder and inside data create db folder.

Then inside

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

Please open the command prompt in bin folder location.

mongod This command is use to start the mongodb database server.

If everything is fine mongo db server start on default port number 27017.

Open another command in same location and run the command mongo : This terminal is use to interact with mongo db database.

To clear the mongo db terminal shell. We need to run the command as

Cntr + L

Mongo db commands

show databases

Or

show dbs

create database databaseName In MySQL

use databaseName;

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

Mongo DB is case sensitive

Create the collection (tables). db is pre defined object which contains lot of pre defined functions to do some operation.

db.createCollection(“collectionName”);

After collection created we can see all collection name present in current database.

Using command as

show collections;

Or

show tables

command to store document (records ) in collection.

db.collectionName.insert({key1:value,key2:value,key3:value});

generally in json key must in double quote. But while storing in mongo db if we want to key in double quote we can pass or without double quote also allow internally it consider as double quote.

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

db.Sample.insert({name:"Ravi"});

View the document from a collection

db.collectionName.find();

whenever we store in document in collection mongo db by default create pre defined property or field ie \_id as primary key which help to maintains the unique ness between two document when document contains same field with same value or different value. This \_id value is use to find the unique ness between two document. In mongo db no primary key \_id is like a primary key.

If we want to pass the custom value we can pass but field name must be \_id. If we pass id or \_ID or empId then internally \_id field create for that document.

Need collection name Emp

\_id name, salary city age

Store five to eight document in Emp collection

In Mongo db we can insert the document in collection without creating. If collection present in it will insert the document in existing collection else it will create and insert.

This query is use to insert many document

db.Emp.insertMany([{\_id:8,name:"Reeta",salary:22000,city:"Mumbai",age:29},{\_id:9,name:"Veeta",salary:23000,city:"Bangalore",age:32}]);

View the specific document from a collection using index position

db.Emp.find()[0]; this query specific complete document in that index positon.

db.Emp.find()[0].name

This query is use to retrieve 0 index position name field value.

Select \* from Emp in RDBMS

Select name,salary from Emp;

db.Emp.find(); in mongo db

To retrieve particular field from a document

db.Emp.find({},{});

First {} is use to write the condition

Second {} is use filter the field.

db.Emp.find({},{name:2}); it display \_id and name field from Emp document

db.Emp.find({},{name:1,age:1}); id display \_id, name, age field from Emp document

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

**Day 6: 11-12-2022**

limit() : this function is use to display number of document from a collection. It is use to display top most documents from a collection.

db.Emp.find().limit(4);

skip() : this function is use to skip the number document from a collection ie top most skip.

db.Emp.find().skip(4);

db.Emp.find().skip(4).limit(2);

sort() : this function is use to sort the document using specific field.

db.Emp.find().sort({salary:1}); 1 means ascending order

db.Emp.find().sort({salary:-1}); -1 means descending order

retrieve the document from a collection using conditions like where clause in RDBMS (filter the documents)

synax

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

db.collectionName.find({condition});

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

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

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

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

db.Emp.find({name:'Ravi'},{age:1,\_id:0});

db.Emp.find({salary:{$gt:20000}});

db.Emp.find({salary:{$gte:20000}});

db.Emp.find({salary:{$lt:20000}});

db.Emp.find({salary:{$lte:20000}});

db.Emp.find({salary:{$eq:20000}});

db.Emp.find({salary:{$ne:20000}});

$and : both the condition satisfies

$or : any of the condition must be satisfies

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

db.Emp.find({$and:[{name:'Ravi'},{salary:{$gt:20000}}]});

db.Emp.find({$or:[{name:'Ravi'},{salary:{$gt:20000}}]});

mongo db provided one of the operator ie $regex (Regular expression)

regular expression is use to search the data base upon the pattern.

db.Emp.find({name:{$regex:'ee'}}); : name contains ee character

db.Emp.find({name:{$regex:'s'}}); : name contains s character

db.Emp.find({name:{$regex:'^R'}}); : start with R character

db.Emp.find({name:{$regex:'a$'}}); : end with a character

db.Emp.find({name:{$regex:'^r'}});

db.Emp.find({name:{$regex:'^r',$options:'i'}}); $option parameter is use to ignore case sensitive.

Update the document fields using some conditions. Like update query in RDBMS

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

db.Emp.update({\_id:1},{$set:{salary:25000}});

db.Emp.update({\_id:2},{$set:{salary:30000,city:"Mysore"}});

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

this is use to update many records with conditions.

$inc operator is use to increment and decrement the field value.

db.Emp.update({\_id:2},{$inc:{salary:500}}); increment the salary

db.Emp.update({\_id:2},{$inc:{salary:-500}}); decrement the salary

update the field value when both the conditions satisfies

db.Emp.update({$and:[{\_id:1},{name:'Ravi'}]},{$inc:{salary:-700}});

**Remove specific document with conditions**

db.collectionName.remove({}) all documents removed

db.Emp.remove({\_id:9})

db.Emp.remove({$and:[{\_id:9},{name:'Reeta'}]});

db.Emp.remove({$or:[{\_id:9},{name:'Reeta'}]});

$exists : this operator is use to check the specific field for all document in a collection

db.Sample.find({age:{$exists:true}});

db.Sample.find({age:{$exists:false}});

$unset and $set

$unset operator is use to remove the field from a documents. It is equal to drop column in RDMBS

db.Sample.update({name:'Balaji'},{$unset:{city:1}});

this query it remove city field for that document where name must be Balaji. It remove only one document if want more then use updateMany

db.Sample.updateMany({},{$unset:{age:1}});

This query it remove age field for all documents. If exists

$set operator

This operator check the conditions if conditions satisfies and the field value replace by new value. If that document doesn’t contains that field then it will that field for that document where the conditions satisfies.

db.Sample.update({name:"Ravi"},{$set:{city:"Mumbai"}});

db.Sample.updateMany({},{$set:{age:21}}) this query is use to add the age field for all document in collection.

We can store array values.

Student

Sid SName Age Phy, Che, Math, Bio, Eng, Hindi Kan Comp

1 Ravi 21 66 54 67 67 56 87

2 null 78

Student

Subject

StudentSubjectRelationship

Storing the array value in collection like Student details.

db.Student.insert({\_id:4,sname:"Mahesh",age:32,result:true,marks:[86,88,67,90]});

db.Student.find({marks:88}); check value 88 marks in any index position

db.Student.find({'marks.0':88}); check value 88 marks in 0 index position

Mongo DB Relationship

Four type of relationship

One to one : Person has one passport or Person has only one address

One to many : Trainer handling more than one students

Many to one : many employee working in one department or in one project.

Many to Many : Employees or students known more than one skillset.

In RDBMS we achieve these relationship using primary key and foreign key with more than one tables.

In Mongo DB we can achieve this relationship using two ways

1. Embedded style relationships (only one collection)
2. Linking style relationship (more than one collection)

Embedded style relationship

db.Employee.insert({\_id:100,name:'Ravi',salary:12000,add:{city:"Bangalore",state:"Kar"}});

db.Employee.insert({\_id:101,name:'Ramesh',salary:14000,add:[{city:"Bangalore",state:"Kar"},{city:"Mumbai",state:"Mh"}]});

db.Employee.insert({\_id:103,name:'Mahesh',salary:18000,add:{city:"Mysore",state:"Kar"},projects:[{pid:111,tech:"Java"},{pid:222:tech:'Node JS'}]});

db.Employee.insert({\_id:103,name:'Mahesh',salary:18000,add:{city:"Mysore",state:"Kar"},projects:[{pid:111,tech:"Java"},{pid:222,tech:'Node JS'}]});

db.Employee.find().pretty();

apply condition for complex property field

db.Employee.find({'add.city':"Mumbai"}).pretty();

db.Employee.find({'projects.tech':"Node JS"}).pretty();

Linking style

We can make separate collection while storing the document in collection in linking style

Trainer

\_id TName tech

1 Raj Java

2 Ravi Python

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

WriteResult({ "nInserted" : 1 })

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

Students1

\_id SName Age tsid (like FK) we are storing tid while storing student document

100 Seeta 21 1

101 Meeta 22 1

102 Reeta 23 2

103 Veeta 24 2

104 Teeta 25 [1,2]

db.Students1.insert({\_id:100,sname:"Seeta",age:21,tsid:db.Trainer.find()[0].\_id});

WriteResult({ "nInserted" : 1 })

> db.Students1.insert({\_id:101,sname:"Meeta",age:22,tsid:db.Trainer.find()[0].\_id});

WriteResult({ "nInserted" : 1 })

> db.Students1.insert({\_id:102,sname:"Reeta",age:23,tsid:db.Trainer.find()[1].\_id});

WriteResult({ "nInserted" : 1 })

> db.Students1.insert({\_id:103,sname:"Veeta",age:24,tsid:db.Trainer.find()[1].\_id});

WriteResult({ "nInserted" : 1 })

> db.Students1.insert({\_id:104,sname:"Teeta",age:24,tsid:[db.Trainer.find()[0].\_id,db.Trainer.find()[1].\_id]});

WriteResult({ "nInserted" : 1 })

Students2

\_id SName Age tsid (like FK) we are storing trainer details while storing student document.

100 Seeta 21 {\_id:1,tname:”Raj”,tech:”Java”}

101 Meeta 22 {\_id:1,tname:”Raj”,tech:”Java”}

102 Reeta 23 {\_id:2,tname:”Ravi”,tech:”Python”}

103 Veeta 24 {\_id:2,tname:”Ravi”,tech:”Python”}

db.Students2.insert({\_id:100,sname:"Seeta",age:21,tsid:db.Trainer.find()[0]});

db.Students2.insert({\_id:101,sname:"Meeta",age:22,tsid:db.Trainer.find()[0]});

db.Students2.insert({\_id:102,sname:"Reeta",age:23,tsid:db.Trainer.find()[1]});

db.Students2.insert({\_id:103,sname:"Veeta",age:24,tsid:db.Trainer.find()[1]});

db.Students2.insert({\_id:104,sname:"Teeta",age:24,tsid:[db.Trainer.find()[0],db.Trainer.find()[1]]});

**Day 7: 17-12-2022**

**if(obj!=null) {**

**this.info = obj;**

**}**

**Aggregate operator : In Aggregate operator or operation, Mongo db process the data records or documents and it return single computed result base upon group.**

**$lookup :this operator we use inside aggregate function to link more than one collection with common field or property to get the result.**

**db.Students1.aggregate([{$lookup:{from:"Trainer",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}}]);**

**db.Trainer.aggregate([{$lookup:{from:"Students1",localField:"\_id",foreignField:"tsid",as:"StudentDetails"}}]);**

**with aggregate we use $group operator**

**db.Employees.insertMany([**

**{\_id:1,name:"Ravi",salary:32000,age:21,city:"Bangalore",deptId:100},**

**{\_id:2,name:"Raju",salary:25000,age:24,city:"Mumbai",deptId:101},**

**{\_id:3,name:"Ramesh",salary:26000,age:22,city:"Delhi",deptId:102},**

**{\_id:4,name:"Rajesh",salary:28000,age:23,city:"Bangalore",deptId:100},**

**{\_id:11,name:"Raghu",salary:14000,age:22,city:"Delhi",deptId:101},**

**{\_id:5,name:"Reeta",salary:12000,age:23,city:"Bangalore",deptId:102},**

**{\_id:6,name:"Ramesh",salary:22000,age:25,city:"Mumbai",deptId:100},**

**{\_id:7,name:"Lokesh",salary:38000,age:28,city:"Mumbai",deptId:102},**

**{\_id:8,name:"Ajay",salary:25000,age:29,city:"Bangalore",deptId:102},**

**{\_id:9,name:"Vijay",salary:35000,age:27,city:"Delhi",deptId:101},**

**{\_id:10,name:"Ram",salary:22000,age:25,city:"Bangalore",deptId:101}**

**])**

**To find total salary for all employees document**

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

**to find maximum salary in employees document**

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

**to find minimum salary in employees document**

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

**to findavg salary in employees document**

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

**Find the total sum salary group by deptId**

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

Find the total sum salary group by city

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

**$match operator : to check the condition with $group operator**

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

Connection mongo db database using Node JS :

Using mongodb module

AWS : EC2 instance : which help to create server machine with we can install all required software to run the application.

In EC2 instance we can run angular application or express js application after installing node js

Once we run any application in EC2 instance (which help to create virtual server machine). That instance provide us unique ip address so we can access express js or angular or mongo db database using <http://ipaddress:42000> or <http://ipaddress:3000> or <http://ipaddress:27017>

Node JS provided external module ie mongodb which provided set of function which help to connect to mongodb database to do operation like insert, delete, update and retrieve documents from a collection.

First create the folder with any name (mongo db program using mongo db module)

Inside this folder create package.json file usng npm init

Install mongodb module using npm command

npm install mongodb

18-12-2022

Express JS with Mongo DB module to connect the mongo db database.

Post

Put

Delete

Get

Employee

Customer

Express JS with MVC Style to connect Mongo db dataset using Mongo db node js module

Model View and Controller

View -🡪 Angular or Express Engine as Jade

Controller and Model layer using Express JS



Create folder with name ProductRestAPI Using Mongodb module

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

npm init (create package.json file) it will ask package name give com. Then hit enter and enter.

Install the module

npm install mognodb

We will create the folder.

config : This folder connect one file which help us to provide the database connection for mongo db. Config (database configuration).

Create the file with name as dbConfig.js

repository : This folder contains more than one file and each file contains more than one function which is responsible to interact with database to do operation like insert, delete, update and retrieve from mongo db or any other database.

This folder contains more than one file with database logic using mongodb or any other module.

Like productRepository

cusomerRepository

mangerRepository

So we create productRepository.js file and we will write more than one function to do some operation on Product collection.

controller : This folder contains more than one file and each file contains function more http methods which response to take request from another file and pass the value to repository layer and base upon response it will pass the value to another as response. Controller layer controller request and response.

Now we will create productController.js file and we will write more than one function which help us to receive the request and pass to repository layer and base upon response it will inform to another layer.

router : this folder contains routing file which is responsible to check subpath of request and as well as http method. Base upon subpath it will pass to controller layer.

This file provide the clear picture of our application all path as well as sub path and controller methods.

To create the router we need express js module

**npm install express**

productRouter.js

app.js (main file)

This file is responsible to load the required module, connect to database using config module, enable any middleware application and running the application on port number.

creating user defined module

if any file contains variable or function or class the scope of the variable, function or class within that file. If we want to access in another file then we have to exports it and in another using require function we have to import it.

app.js ---🡪 DbConfig.js ---🡪productRouter.js -🡪productController.js --🡪productRepository.js(using dbConfig.js)---🡪 calling mongo db database to do operation.

dbConfig.js ( this file contains two function

connection and CollectionInformation

connection function responsible to provide the connection

CollectionInformation function take two parameter ie 1st is database name and 2nd parameter is collection name.

productRepository.js This file using require function load the dbConfig.js file and call the CollectionInformation function to get the database details and collection name and it will do the operation like insert, delete, update and retrieve.

This file must contains only pure database logic.

Linking one file and another file using require function to load the module in JavaScript ES5 style.

From JavaScript ES6 as well as Typescript we use import rather than require.

productController.js : this file contains more than one function each function take two parameter. Request and response. Request is use to receive the request from client and response is use to give the response back to client. These function receive request from client and pass the data to repository layer.

Using await keyword we are making asynchronous statement as asynchronous.

If we use await keyword inside a function for that function we need to use async keyword.

productRouter.js

Inside this file we have to create the reference of router with help of express module and this reference provide get, post, put, patch and delete method which take two parameter 1st parameter is subpath of request and 2nd parameter is controller reference with function name.

App.js ( it is main file) which load all required module like express, then add middleware module like enable json or another module. Then check the main path of the application and add middleware for router module and run the application on specific port number.

07/01/2022

Product CRUD Operation

1. Get all products details : Get Method

http://localhost:3000/api/products/findProductInfo

1. Search product using productId : Get method

http://localhost:3000/api/products/findProductById/1

1. Store the product details

http://localhost:3000/api/products/storeProduct

node js provided external module ie nodemon which help to refresh our node js project if we do any changes in project.

pm install nodemon –g

or

npm install nodemon

nodemon app.js rather than node app.js

After created all rest API using express module

Now create the angular project

ng new product-angular-app

routing -🡪 yes

styling 🡪 css

ng g c product (create the component)

ng g s product (create the service)

ng g class product (create the model class which help to map the json data)

to call RESTPI we need to do DI for HttpClient in service class.

HttpClient API is part of HttpClientModule. So we need to import HttpClientModule in app.module.ts file in imports section.

Angular forms

Template Driven Form : ngForm and ngModel attribute

Model Driven Form or Reactive forms :FormGroup and FormControl

If we are planning to use Template driven form then we need to import FormsModule and if we are planning use reactive form then we need to import ReactiveFormsModule in app.module.ts file in import section.

CORS policy

CORS : Cross Origin Resource Sharing

Angular project -🡪 internal web server with default port number 4200

Express JS 🡪 running on node js server with port number 3000

Two domain or server are communicate to each others. So browser disable the features ie cors.

To enable cors policy using node js. Node js provided external module ie cors. So we need to install cors module and add as middleware to enable cors policy using express js.

npm install cors

08-01-2023

Mongoose : mongoose is a another external module provided by node js which help to connect the database like mongo db.

Mongo is known a native api driver which help do the operation with mongo db database.

Mongoose is provided object modeling tool which help to control the collection structure like name of the collection, number of attribute as well as value inside a attribute. OMT provide the structure for the collection.

Mongo db like a JDBC

Mongoose is like ORM (Object Relation Mapping)

Mongo db is a database

To connect mongo db database using node js or express js we can use two module

Ie mongodb module as well as mongoose module

Create the folder

**StudentRestAPI Using Mongoose module – backend**

Plz open this folder in vs code.

Using npm init create package.json file

Install two modules

Express and mongoose

**npm install express mongoose**

config : this folder contains file which is responsible to provide db connection

dbConfig.js

model : this folder contains set of files which provide the structure for the collection like Student is collection name which contains firstname, lastname,age,marks fields, firstname is mandatory with string type, last name is optional, age is number and marks number.

studentModel.js

repository : this folder contains set of files which contains more than one function which is responsible to do the operation on database.

studentRepository.js

controller this folder contains set of files which is responsible to take the request from router file and pass the repository base upon the result will send the message to client application.

router : this file contains set of files which is responsible to check the sub path of request and base upon request it will pass the flow of the application to particular function or method of controller layer.

app.js : this is entry of the application.