Date: 6/1/18

Topic: (To do Rest APP in Express)

In this tutorial we are going to learn backend part and we will use (node , express, and mysql) and later we will connect one more technology which is React JS.

So I think everything should be setup before going to move on the project.

So basically you need to install the latest version of nodejs and then we will create the express js applicaton and we will learn how to integrate the mysql data into out express application.

* Set up the express server
* And fallow the below intsrction.

So for each task we will follow the mvc architecture but not complete “MVC” so for each and every taksk we will define the route and for specific route we will define the controller.

So lets go how we will do

---------------------------------

* Create an express folder as “express\_todo\_app”
* And navigate that folder through the command
* Now initialize the node modules in it “npm init”
* Fill the require instructions.
* Now install the express module “npm install express –save”
* You will get the new folder of “node\_modules”
* Now create a new file “app.js” and this file will be the entry point of our project .

Code “app.js”

// load the express module

const express = require('express');

// create express app instances

var app = express();

// express server

var port = 5000;

app.listen(port,() =>{

console.log("Server running on Port No :"+port);

})

Second Step-

Install the mysql database through the “npm”

Use the command “npm intall mysql --save”

And create a connection file name as “database.js”

Befor that create one new folder “app” inside that folder create>and create another folder “lib”>databse.js

The file contains the following code –

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "root",

password: ""

});

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

});

// export the connection to use it anohter file

module.exports = con;

Now link this database file inside your app.js file

// import the mysql connection

var con = require('./app/lib/database');

now create some end points for the task

till now we have done –

we have established the database connection and basic express app structure.

So now for

----------------

We will create one table in my databse (remember that in connection code we have not specified te database name so for that we will specify and we will create one table )

Column name should be

Column name -------------Properties

Id integer,autoincrement,primary key

First\_name text

Last\_name text

Email\_id text

Date\_of\_birth DateTime

Notes:

To get the body request we need to install the dependencies of “body-parser”

So go to your npm command and istall this dependencies using the command of “npm install bodyparser”

Databse name is “nodemysql” and table name is “student”

So if you have install the body parser then you have to use some middle ware to parse the body data

So apply this syntax in the “app.js” file

First import the body parser and then use it in the middleware

// import the body parser module

var bodyParser = require('body-parser');

// create express app instances

var app = express();

// apply middle ware like

app.use(bodyParser.urlencoded({extended:true}));

app.use(bodyParser.json());

now you have setup the bodyparser middleware

now make a post route to check it is working or not

so in “app.js” file create one post route the syntax is look like as –

// load the express module

const express = require('express');

// import the mysql connection

var con = require('./app/lib/database');

// import the body parser module

var bodyParser = require('body-parser');

// import the controller

var controller = require('./app/controller/studentTask');

// create express app instances

var app = express();

// apply middle ware like

app.use(bodyParser.urlencoded({extended:true}));

app.use(bodyParser.json());

// define the post route

app.post('/savedata',controller.saveStudentData);

// express server

var port = 5000;

app.listen(port,() =>{

console.log("Server running on Port No :"+port);

})

App>controller>studentTask.js

module.exports = {

saveStudentData : function(req, res){

// access the data in the request body

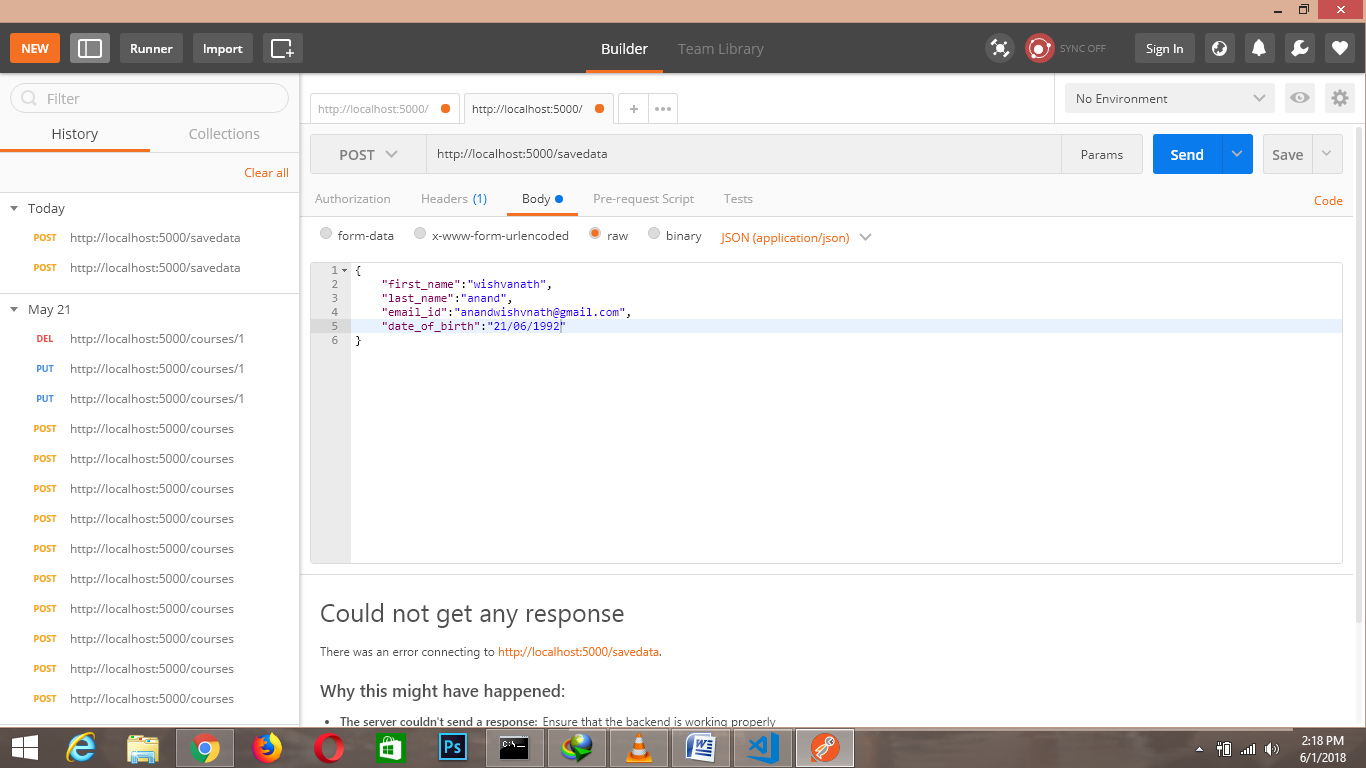
var requestData = req.body;

console.log(requestData);

}

};

Now open postman application in your chrome browser and send the valid request you will see the response generated by body in console.

The output is –

{ first\_name: 'wishvanath',

last\_name: 'anand',

email\_id: 'anandwishvnath@gmail.com',

date\_of\_birth: '21/06/1992' }

in your console.

So follow the below code and understand how you should write the database logic

studentTask.js –

// import the mysql connection object

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

module.exports = {

saveStudentData : function(req, res){

// access the data in the request body

var requestData = req.body;

console.log(requestData);

// if data is not available return some error message

if(!(requestData.firstname||requestData.lastname||requestData.email\_id)){

return res.status(400).json({ErrorCode:"Your haven't mentioned all the data"})

}

// create a variable to store the data which is coming through the request body

var studentData ={

first\_name: requestData.first\_name,

last\_name: requestData.last\_name,

email\_id: requestData.email\_id,

date\_of\_birth: requestData.date\_of\_birth

};

// write the query

con.query("INSERT INTO `student` SET ?",studentData,function(err,request,fields){

if(err){

console.log(err);

return res.status(500).json({ErrorCode:"Error Occured to insert the student data"});

}

return res.status(200).json({Code:"Student data has been successfuly Saved"});

});

}

};

Now post the some valid data by the postman

And check your table into the database you will get one new row which you have posted earlier now through the postman.