|  |
| --- |
| var express = require("express"); |
|  | var app = express(); |
|  | var port = 3000; |
|  | var bodyParser = require('body-parser'); |
|  | app.use(bodyParser.json()); |
|  | app.use(bodyParser.urlencoded({ extended: true })); |
|  |  |
|  | var mongoose = require("mongoose"); |
|  | mongoose.Promise = global.Promise; |
|  | mongoose.connect("mongodb://localhost:27017/node-demo"); |
|  | var nameSchema = new mongoose.Schema({ |
|  | firstName: String, |
|  | lastName: String |
|  | }); |
|  | var User = mongoose.model("User", nameSchema); |
|  |  |
|  | app.get("/", (req, res) => { |
|  | res.sendFile(\_\_dirname + "/index.html"); |
|  | }); |
|  |  |
|  | app.post("/addname", (req, res) => { |
|  | var myData = new User(req.body); |
|  | myData.save() |
|  | .then(item => { |
|  | res.send("Name saved to database"); |
|  | }) |
|  | .catch(err => { |
|  | res.status(400).send("Unable to save to database"); |
|  | }); |
|  | }); |
|  |  |
|  | app.listen(port, () => { |
|  | console.log("Server listening on port " + port); |
|  | }); |

What I had in mongodb-student I took off to try code like up above. And this was under

“ var bodyParser = require(‘body-parser);”

//var routes = require('./routes/student');

var session = require('express-session');

var mongodb = require('mongodb');

var mocha = require('mocha');

var assert = require('assert');

var path = require('path');

var dbConn = mongodb.MongoClient.connect('mongodb://localhost:27017/student/studentinfos', { useUnifiedTopology: true });

var assert = require('assert');

var urlencodedParser = bodyParser.urlencoded({ extended: false});

This is what I changed it too now going back

var express = require('express');

var app = express();

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

app.use(bodyParser.json());

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

var count = 0;

var StudentInfo = require('./models/studentInfo');

var mongoose = require('mongoose');

mongoose.Promise = global.Promise;

mongoose.connect('mongodb://localhost:27017/node-demo', {useNewUrlParser: true},{ useUnifiedTopology: true });

//ES6 Promise stops the "DeprecationWarning" by installing this library

app.set('view engine', 'ejs');

app.use('/assets', express.static('assets'));//This is middleware to use or get styles.css code to be used

//app.use('/student',routes);

app.get('/', function(req, res){

var data = {qs: req.query, count: count};

res.sendFile(\_\_dirname + '/index.ejs', {qs: req.query, count: count});

});

app.post("/addname", (req, res) => {

var myData = new User(req.body);

myData.save()

.then(item => {

res.send("Name saved to database");

})

.catch(err => {

res.status(400).send("Unable to save to database");

});

});

app.post('/addname', urlencodedParser, function(req, res){

console.log(req.body)

var stud = new StudentInfo({

firstName: req.body.firstname,

lastName: req.body.lastname

});

stud.save().then(function(){

assert(stud.isNew === false);

});

count++;

res.render('index-success', {data: req.query});

});

//Get home page

app.post('/views/index-success', function(req, res){

dbConn.then(function(db){

delete req.body.\_id;//for safety reasons

db.collection('feedbacks').insertOne(req.body);

});

res.send('Data recieved:\n' + JSON.stringify(req.body));

res.render('index', {qs: req.query, count: count});

});

app.get('/views/addedStudentPage', function(req, res){

res.render('addedStudentPage', {data: req.query});

});

app.get('/views/savedStudents', function(req, res){

res.render('savedStudents', {data: req.query});

});

// app.post('views/savedStudents', function(){

//

// });

app.get('/views/index-success', function(req, res){

// dbConn.then(function(db){

// db.collection('feedbacks').find({}).toArray().then(function(feedbacks){

// res.status(200).json(feedbacks);

res.render('index-success', {data: req.query});

// });

});

app.use(function(req, res, next){

if(req.method == "POST"){

var count = req.session.count;

console.log(count);

}

next();

});

app.get('/', (req, res, next) => {

res.sendStatus(200);

next();

});

app.listen(process.env.port || 4000, function(){

console.log('app started!');

console.log('listening for requests');

});

From index.ejs under lastName

<input type="text" placeholder="Degree" name="department" value="<%= qs.dept %>" ><br>

<input type="text" placeholder="Program" name="program" >