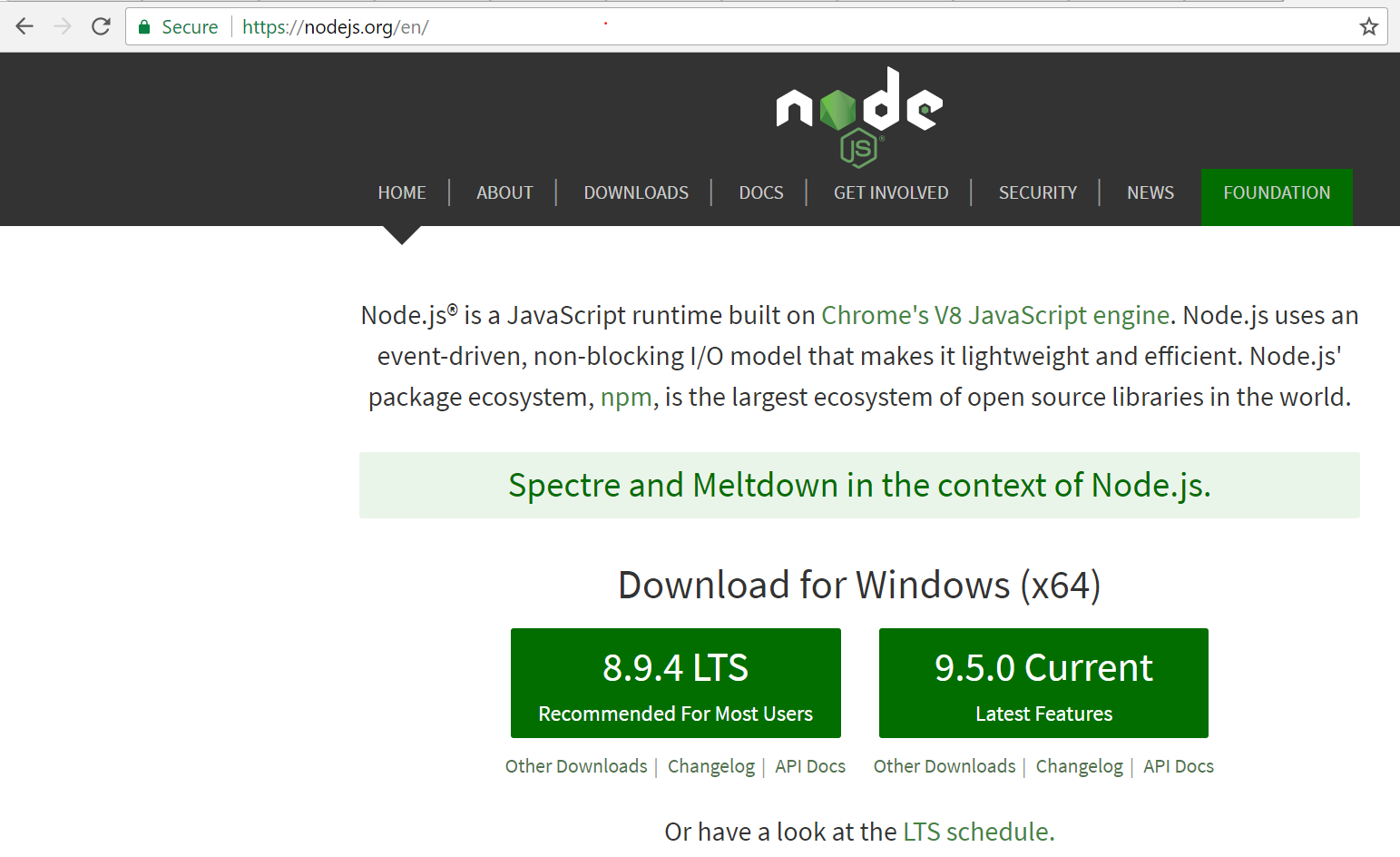
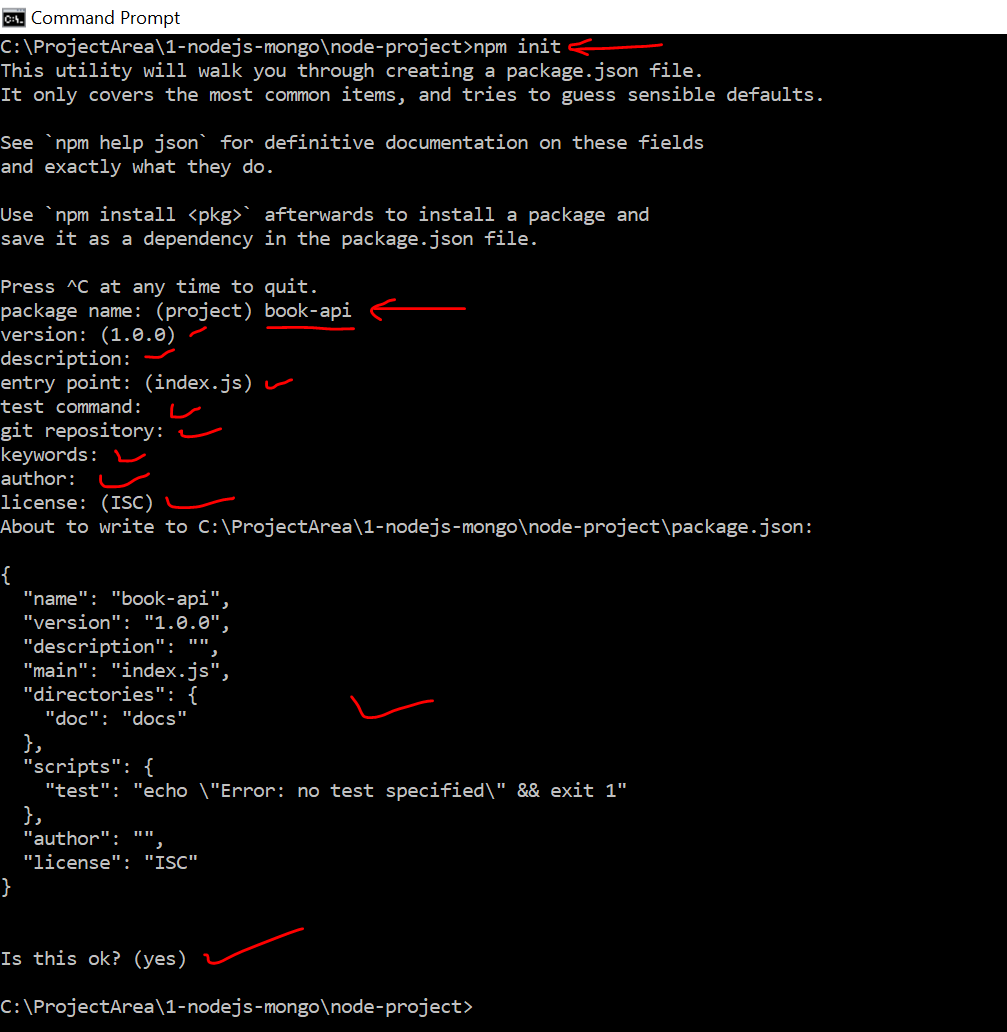
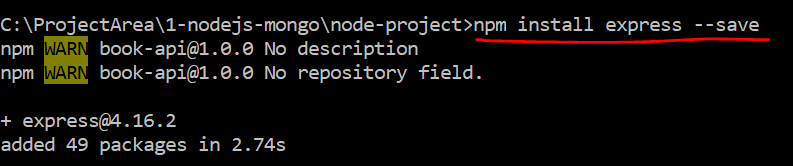
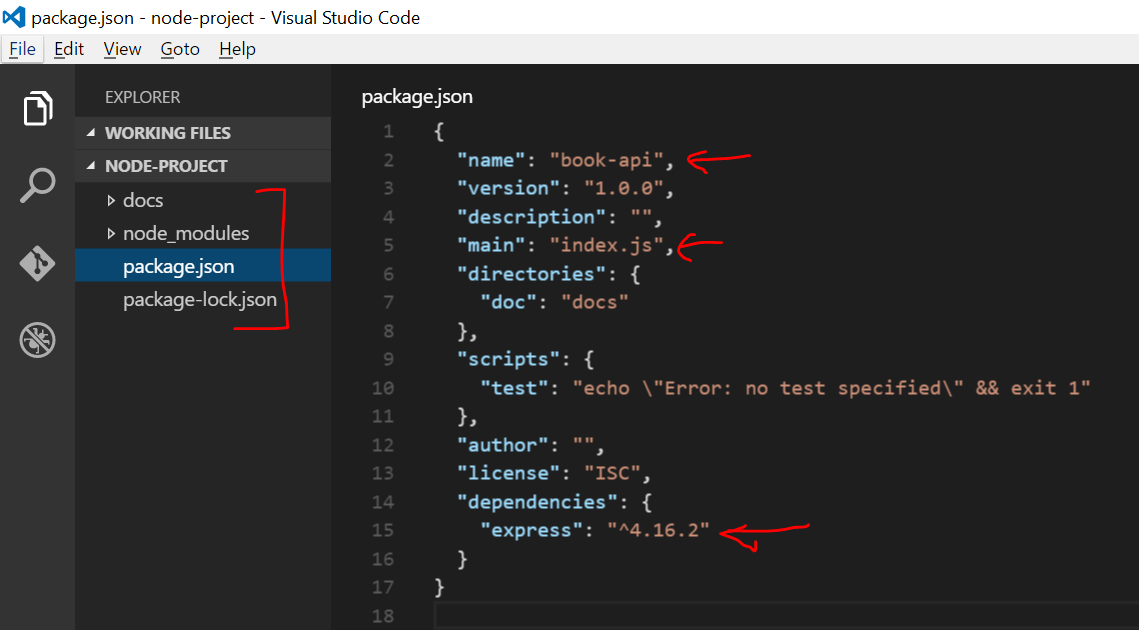
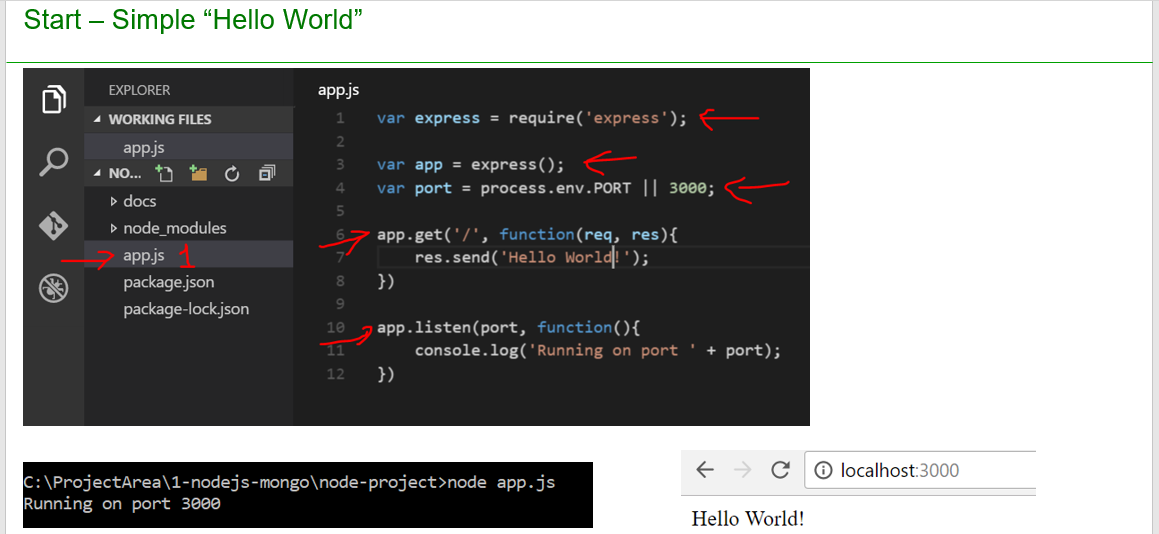
Installation

1. STEP1 – Install node from nodejs.org
   1. 
2. **STEP3 – npm init from any project folder**
   1. 
3. **STEP3 – Install Express (Notice package.json after the command)**
   1. 
4. **STEP4 – Open the code is editor and notice..** 
   1. 
5. Add app.js file at root level manually if not exists
6. Simple ‘Hello World’
   1. 
   2. Open app.js and past the below code

var express = require('express')

var app = new express()

var port = 3000

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

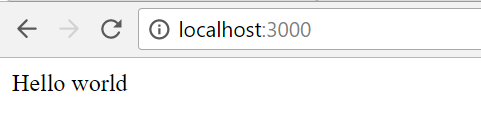
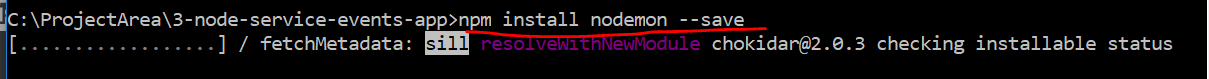
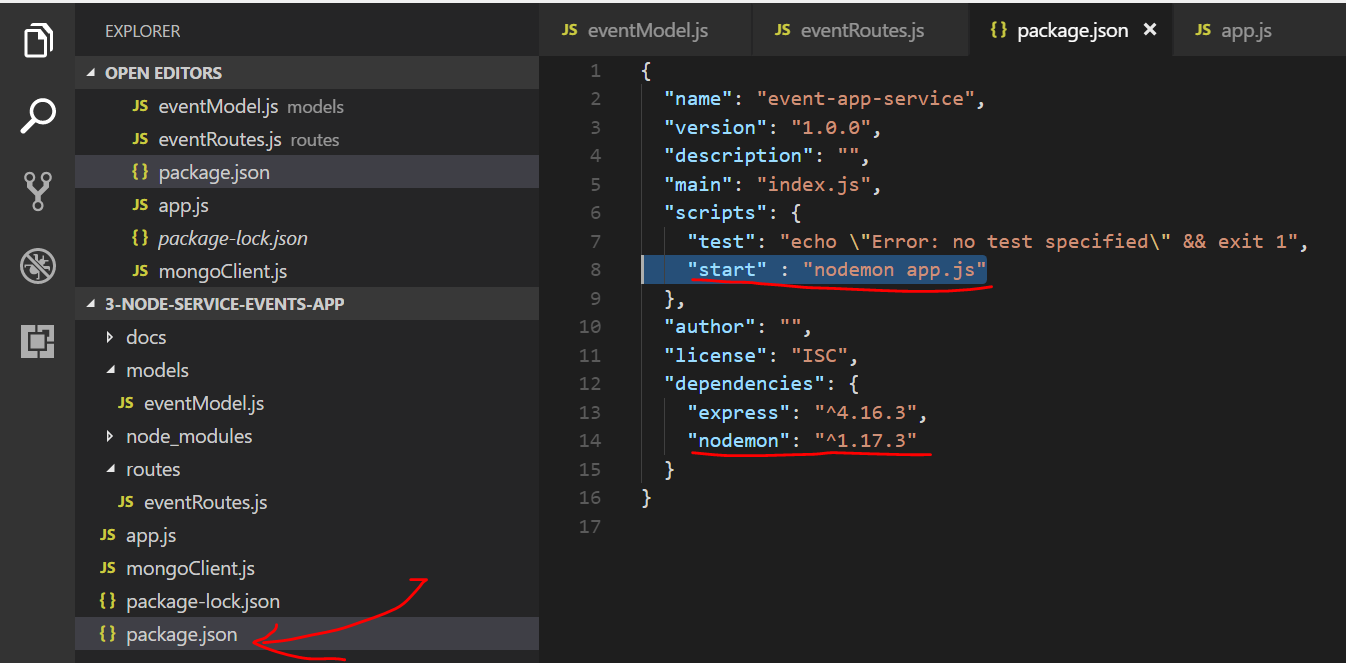
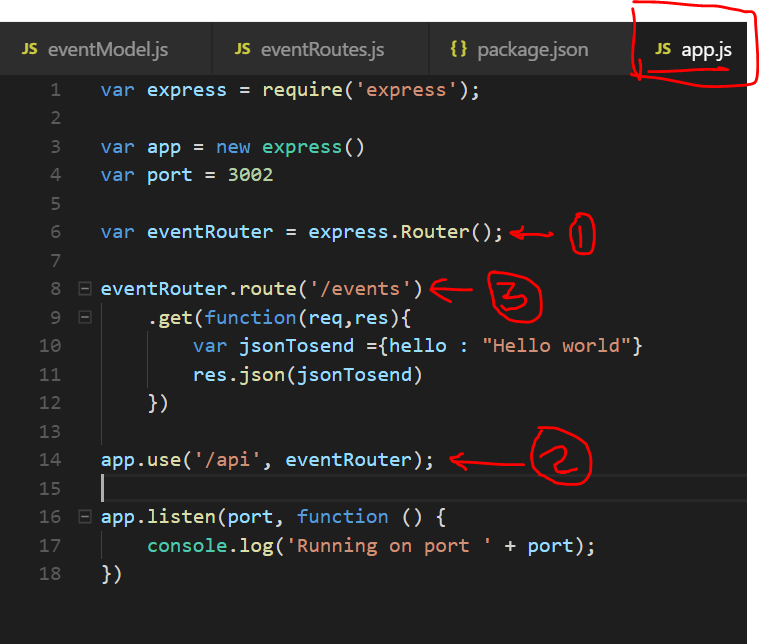
res.send('Hello world')

})

app.listen(port,function(){

console.log('Running on port' + port)

})

1. Go to project folder and run below command
   * 1. Node app.js
2. Type localhost:3000 is chrome(browser) to get the message
   1. 
3. Nodemon – Install nodemon to rerun the node server every time code is changed
   1. 
   2. Add the highlighted line to package.json as shown below
   3. 
4. GET API using router
   1. 
   2. Paste the below code in app.js and run <http://localhost:3002/api/events>

var express = require('express');

var app = new express()

var port = 3002

var eventRouter = express.Router();

eventRouter.route('/events')

.get(function(req,res){

var jsonTosend ={hello : "Hello world"}

res.json(jsonTosend)

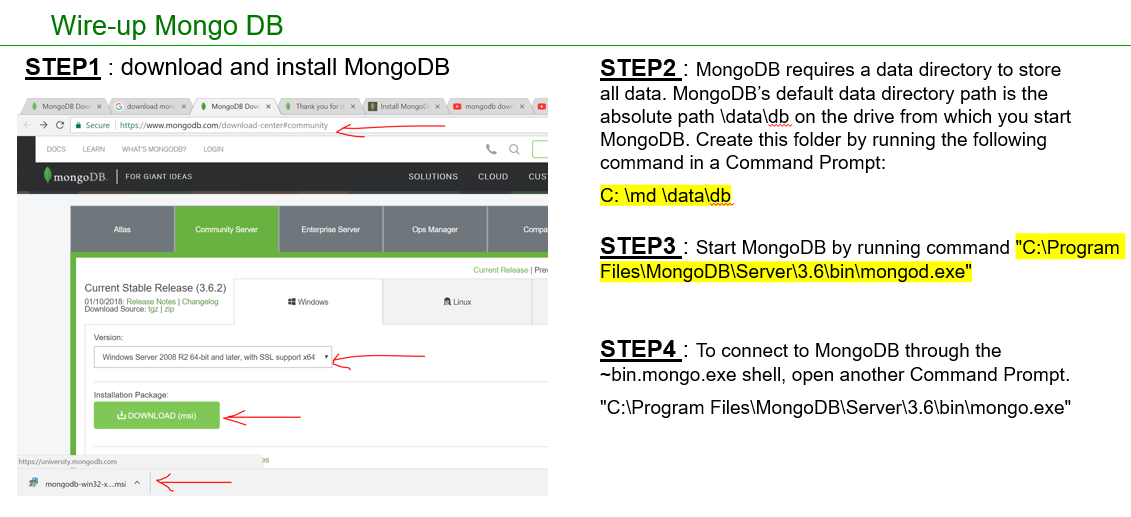
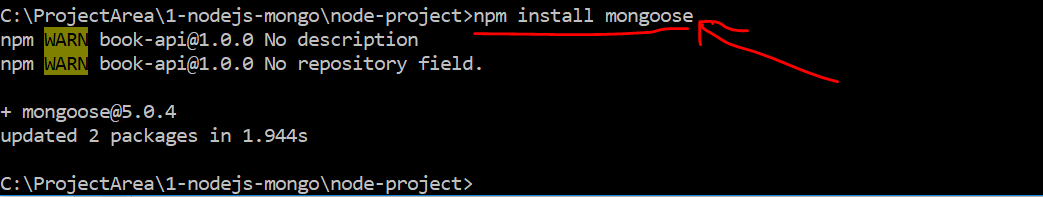
})

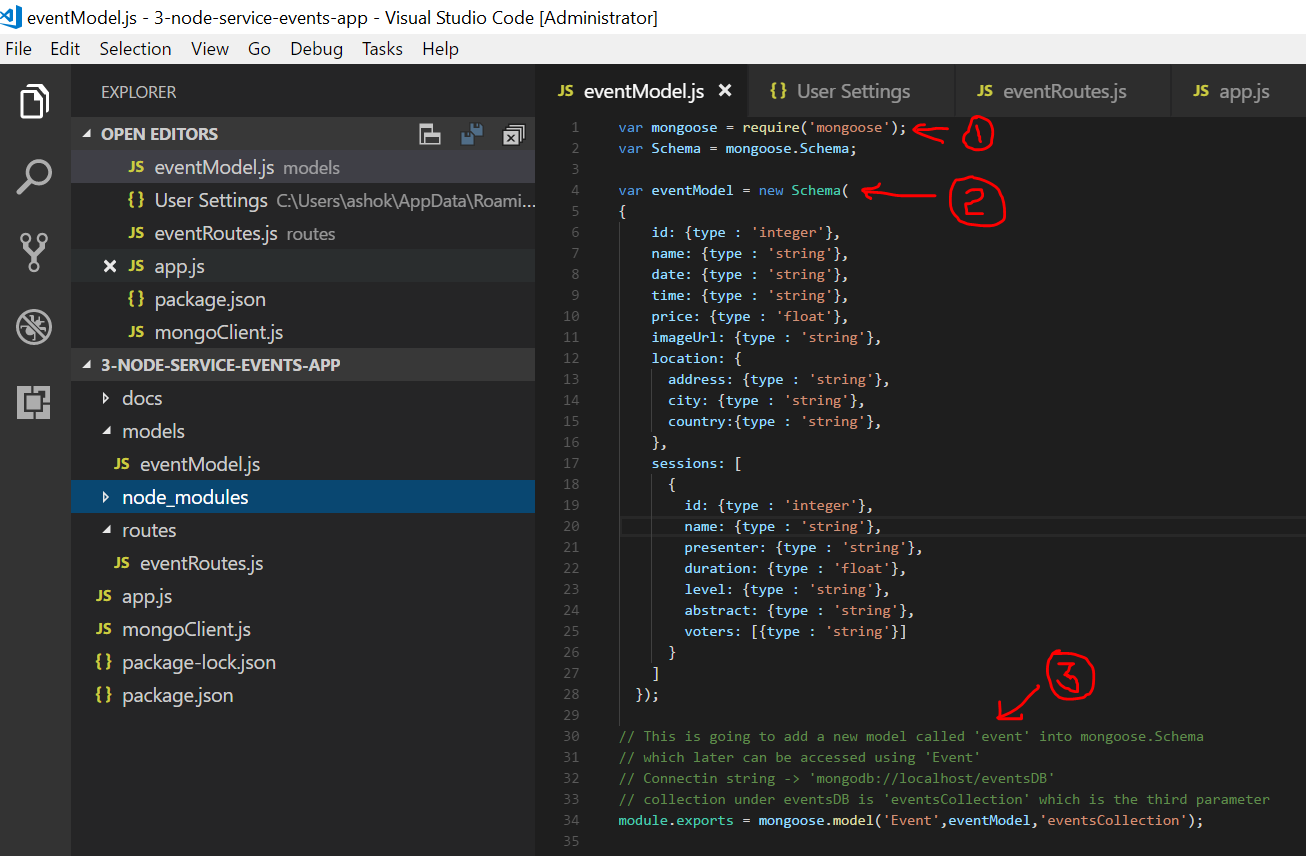
app.use('/api', eventRouter);

app.listen(port, function () {

console.log('Running on port ' + port);

})

1. Wire up MongoDB
   1. 
2. GET API using Events model and mongoose
   1. Install mongoose using npm install mongoose
   2. 
   3. Add a file eventModel.js and paste below code



var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var eventModel = new Schema(

{

id: {type : 'number'},

name: {type : 'string'},

date: {type : 'date'},

time: {type : 'string'},

price: {type : 'number'},

imageUrl: {type : 'string'},

location: {

address: {type : 'string'},

city: {type : 'string'},

country:{type : 'string'},

},

sessions: [

{

id: {type : 'number'},

name: {type : 'string'},

presenter: {type : 'string'},

duration: {type : 'number'},

level: {type : 'string'},

abstract: {type : 'string'},

voters: []

}

]

});

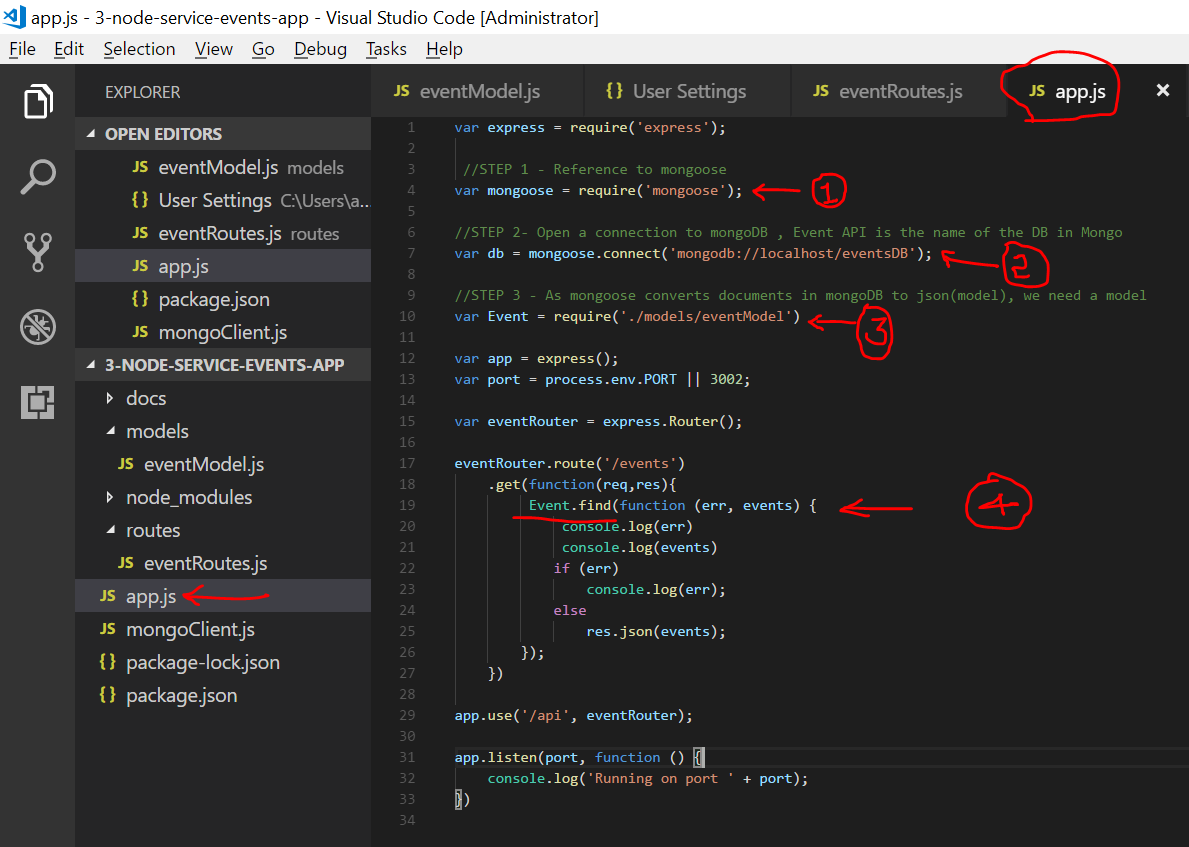
// This is going to add a new model called 'event' into mongoose.Schema

// which later can be accessed using 'Event'

// Connectin string -> 'mongodb://localhost/eventsDB'

// collection under eventsDB is 'eventsCollection' which is the third parameter

module.exports = mongoose.model('Event',eventModel,'eventsCollection');

* 1. Update app.js with below code
  2. 

var express = require('express');

//STEP 1 - Reference to mongoose

var mongoose = require('mongoose');

//STEP 2- Open a connection to mongoDB , Event API is the name of the DB in Mongo

var db = mongoose.connect('mongodb://localhost/eventsDB');

//STEP 3 - As mongoose converts documents in mongoDB to json(model), we need a model

var Event = require('./models/eventModel')

var app = express();

var port = process.env.PORT || 3002;

var eventRouter = express.Router();

eventRouter.route('/events')

.get(function(req,res){

Event.find(function (err, events) {

console.log(err)

console.log(events)

if (err)

console.log(err);

else

res.json(events);

});

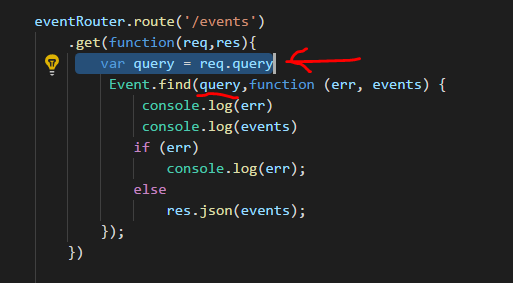
})

app.use('/api', eventRouter);

app.listen(port, function () {

console.log('Running on port ' + port);

})

1. GET API with filtering
   1. 
   2. Place the below code appropriately in app.js ( do not overwrite entire app.js)

eventRouter.route('/events')

.get(function(req,res){

var query = req.query

Event.find(query,function (err, events) {

console.log(err)

console.log(events)

if (err)

console.log(err);

else

res.json(events);

});

})

1. GET API to get a single item
   1. Place the below code appropriately in app.js ( do not overwrite entire app.js) and call as <http://localhost:3002/api/events/5add18a3fb6dcf9c3de84f9a> where 5add18a3fb6dcf9c3de84f9a is the unique id generated by mongodb for the record

eventRouter.route('/events/:Id')

.get(function(req,res){

Event.findById(req.params.Id,function (err, event) {

if (err)

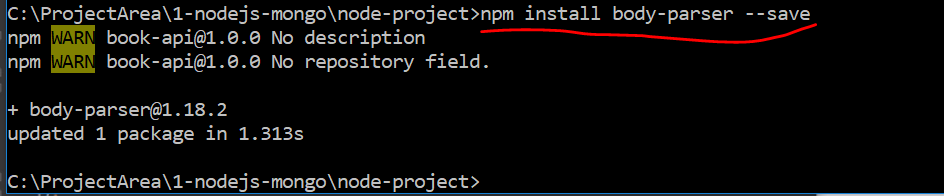
res.status(500).send(err);

else

res.json(event);

});

})

1. Posting data
   1. Install body parser package which is required to read the posted data and convert them to json
   2. 
   3. POST API code changes in app.js
      1. Place the below code appropriately in app.js ( do not overwrite entire app.js)

var bodyParser = require('body-parser'); // Reference to bodyparser

// We have to tell app that we are going to use body parser

//app.use(bodyParser) // Not this

// we have to explicitly tell we are using json parser // support parsing of application/json type post data

app.use(bodyParser.json())

//support parsing of application/x-www-form-urlencoded post data

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

eventRouter.route('/events')

.get(function(req,res){

var query = req.query

Event.find(query,function (err, events) {

console.log(err)

console.log(events)

if (err)

console.log(err);

else

res.json(events);

});

})

.post(function(req,res){

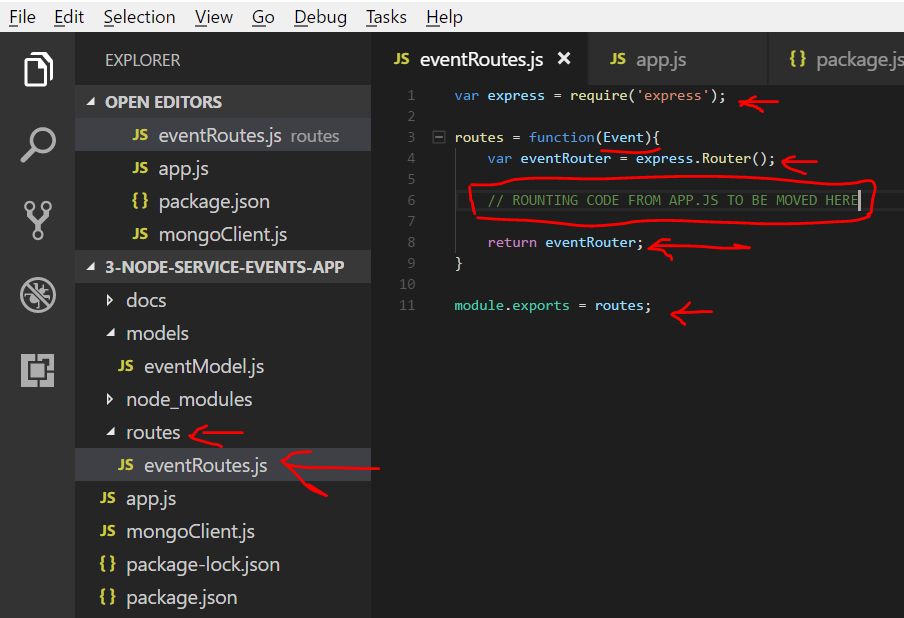
var event = new Event(req.body)

event.save()

res.status(201).send(event)

})

* 1. Test the POST API in SOAPUI or POSTMAN tool using below json data
     1. 

1. Injecting model with code cleanup
   1. As our code in app.js is growing we need move some code(routing API code) to a separate file and inject books into it
   2. Create a file as shown below
   3. 
   4. Place the below code appropriately in eventRoutes.js

var express = require('express');

routes = function(Event){

var eventRouter = express.Router();

// ROUNTING CODE FROM APP.JS TO BE MOVED HERE

eventRouter.route('/events')

.get(function(req,res){

var query = req.query

Event.find(query,function (err, events) {

console.log(err)

console.log(events)

if (err)

console.log(err);

else

res.json(events);

});

})

.post(function(req,res){

var event = new Event(req.body)

event.save()

res.status(201).send(event)

})

eventRouter.route('/events/:Id')

.get(function(req,res){

Event.findById(req.params.Id,function (err, book) {

if (err)

res.status(500).send(err);

else

res.json(book);

});

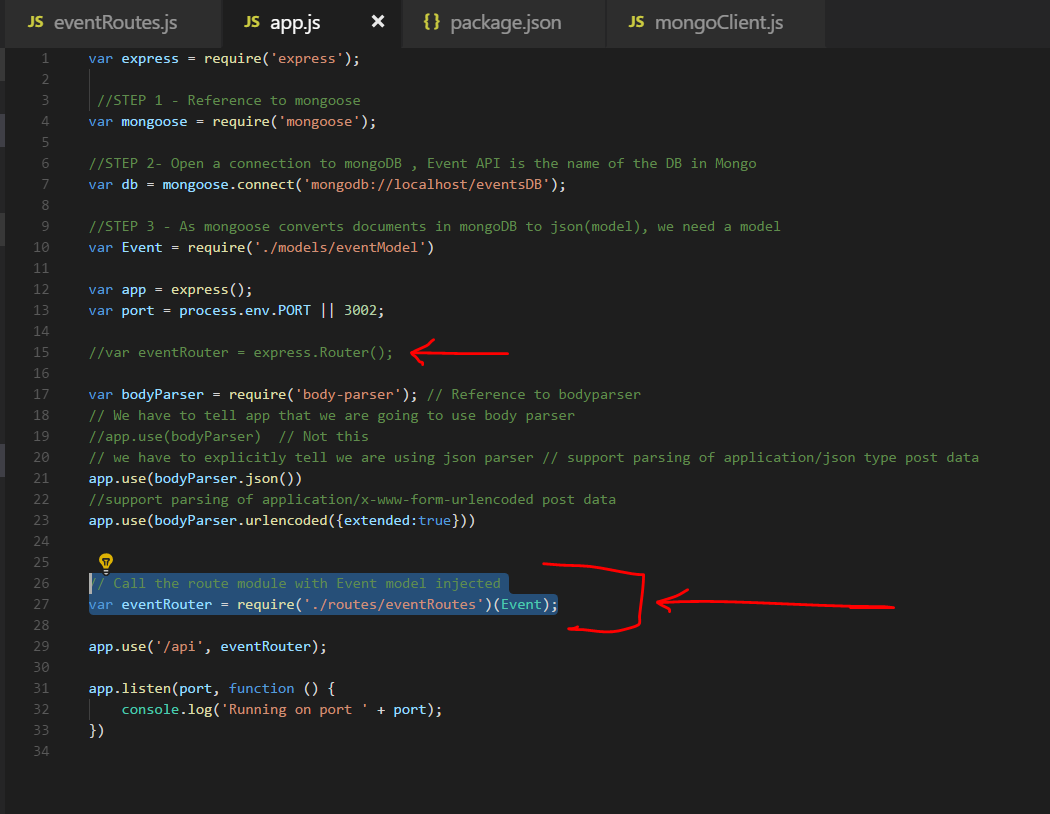
})

return eventRouter;

}

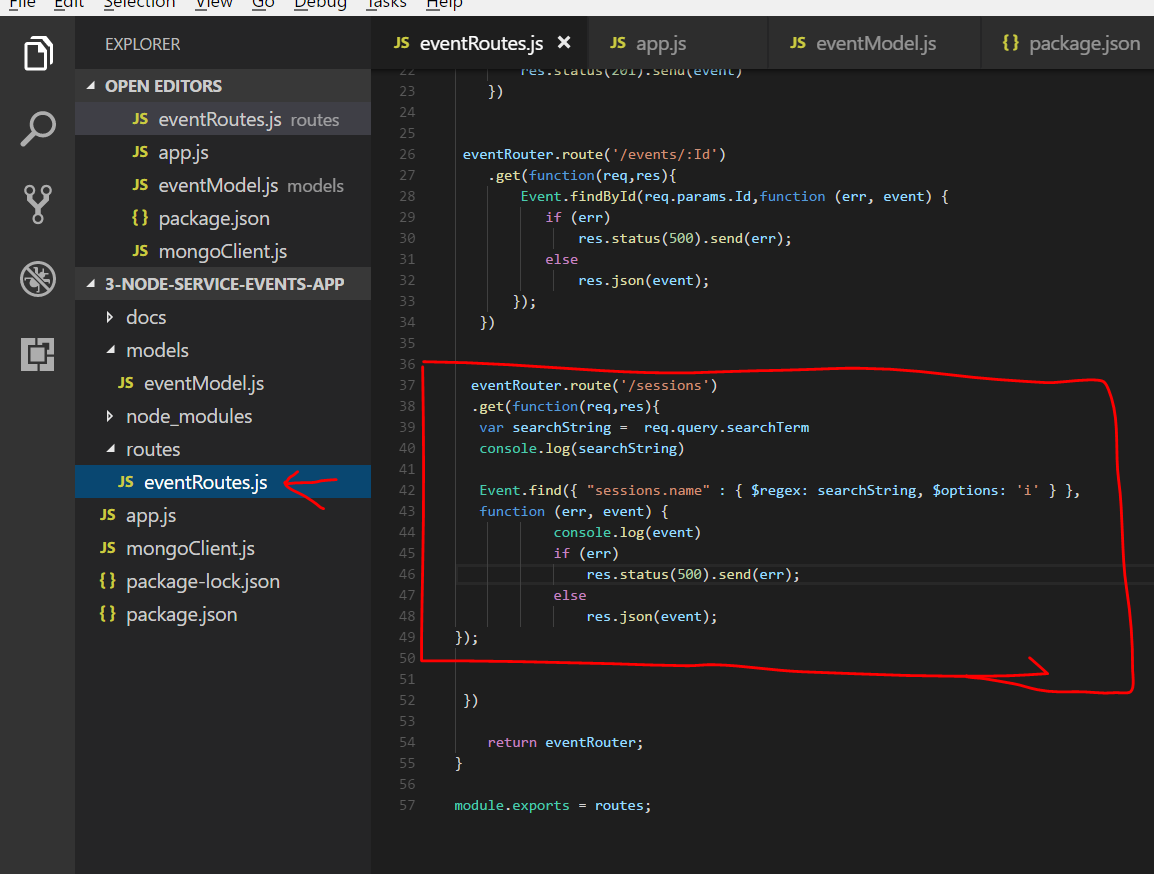
module.exports = routes;

1. Place the below code appropriately in app.js ( do not overwrite entire app.js)



// Call the route module with Event model injected

var eventRouter = require('./routes/eventRoutes')(Event);

1. Get API with search
   1. Add below code in evenRoutes.js
   2. 

eventRouter.route('/sessions')

.get(function(req,res){

var searchString = req.query.searchTerm

console.log(searchString)

Event.find({ "sessions.name" : { $regex: searchString, $options: 'i' } },

function (err, event) {

console.log(event)

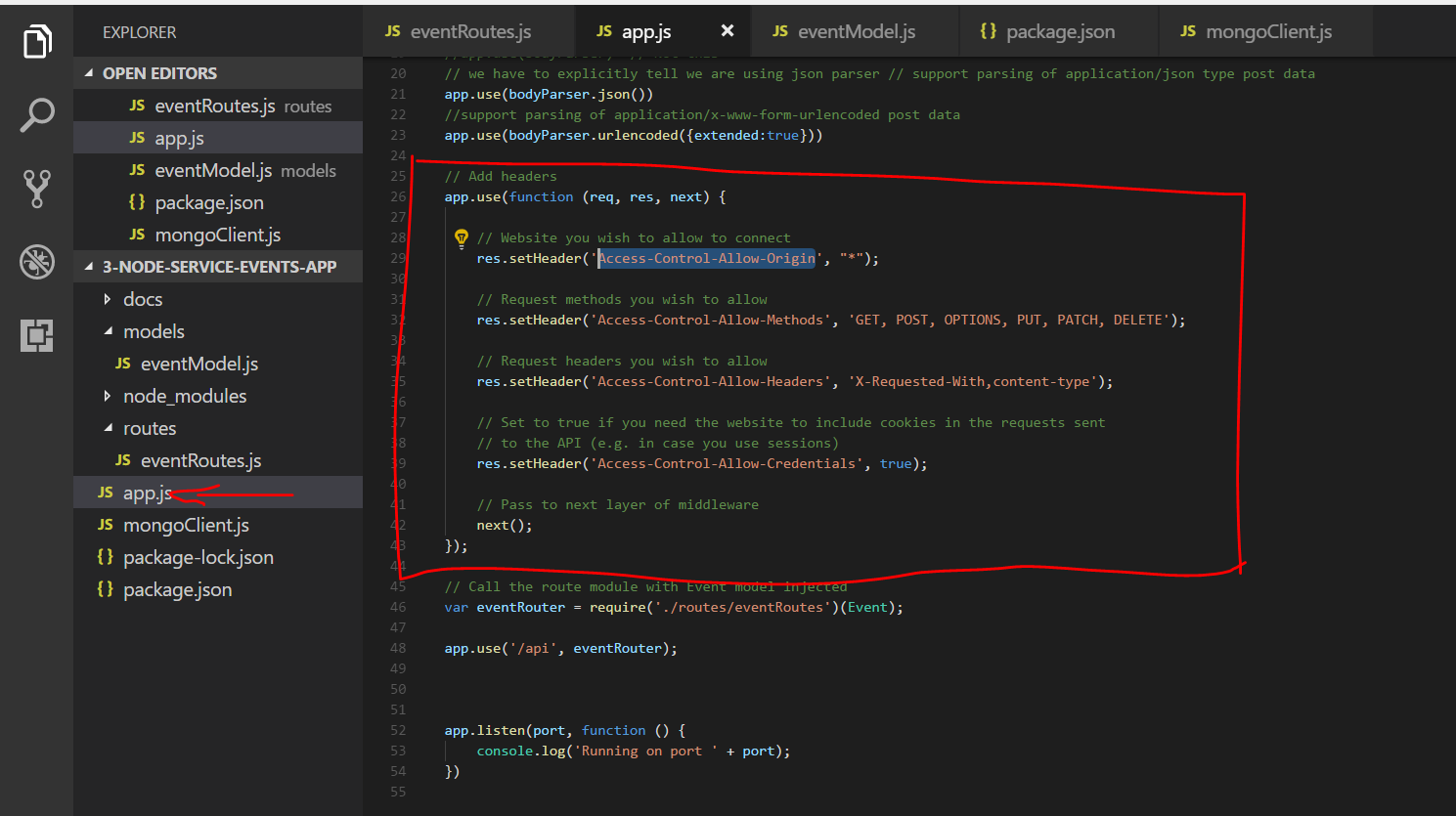
if (err)

res.status(500).send(err);

else

res.json(event);

});

1. Resolve “Access-Control-Allow-Origin” issue
   1. Paste the below code appropriately as show below
   2. 

// Add headers

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

// Website you wish to allow to connect

res.setHeader('Access-Control-Allow-Origin', "\*");

// Request methods you wish to allow

res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH, DELETE');

// Request headers you wish to allow

res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type');

// Set to true if you need the website to include cookies in the requests sent

// to the API (e.g. in case you use sessions)

res.setHeader('Access-Control-Allow-Credentials', true);

// Pass to next layer of middleware

next();

});