|  |  |  |
| --- | --- | --- |
| NODE JS API | | |
|  |  |  |
| Startup Checklist | | |

# INSTALL

* npm i express
* npm i mongoose
* npm i body-parser

# GET API

1. create models folder
2. create model (getAPI.js)
3. write code in getAPI.js

const mongoose = require('mongoose');

let userSchema = new mongoose.Schema({

    \_id:mongoose.Schema.Types.ObjectId,

    name:String,

    address:String,

    email:String

});

module.exports=mongoose.model('users',userSchema);//table name

4.create app.js

//for GET API

const express = require('express');

const app = express();

const mongoose = require('mongoose');

const User = require('./models/getAPI');//model

mongoose.connect('mongodb+srv://brijesh:bkrk0403@cluster1.8qcjw.mongodb.net/demo?retryWrites=true&w=majority',

    {

        useNewUrlParser: true,

        useUnifiedTopology: true

    }

);

// for all record

// app.get('/users', (req,res)=>{

//    User.find().then((data)=>{ //then is promise

//         res.json(data)

//    })

// })

//for specific record

app.get('/users', (req,res)=>{

    User.find().select('name').then((data)=>{ //select field name like name ,age

         res.json(data)

    })

 })

app.listen(4000)

# POST API

1. app.js

const express = require('express');

const app = express();

const mongoose = require('mongoose');

var bodyparder = require('body-parser'); // for post api

var jsonparser = bodyparder.json();

const User = require('./models/getAPI');//model

mongoose.connect('mongodb+srv://brijesh:bkrk0403@cluster1.8qcjw.mongodb.net/demo?retryWrites=true&w=majority',

    {

        useNewUrlParser: true,

        useUnifiedTopology: true

    }

);

// for GET API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// for all record

// app.get('/users', (req,res)=>{

//    User.find().then((data)=>{ //then is promise

//         res.json(data)

//    })

// })

//for specific record

// app.get('/users', (req,res)=>{

//     User.find().select('name').then((data)=>{ //select field name like name ,age

//          res.json(data)

//     })

//  })

// for post API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

app.post('/user', jsonparser,(req,res)=>{ // 2nd parameter is middleware

    const data = new User({

        \_id:new mongoose.Types.ObjectId(),

         name:req.body.name,

          address:req.body.address,

          email:req.body.email

    })

       data.save().then((result)=>{ //then is promise

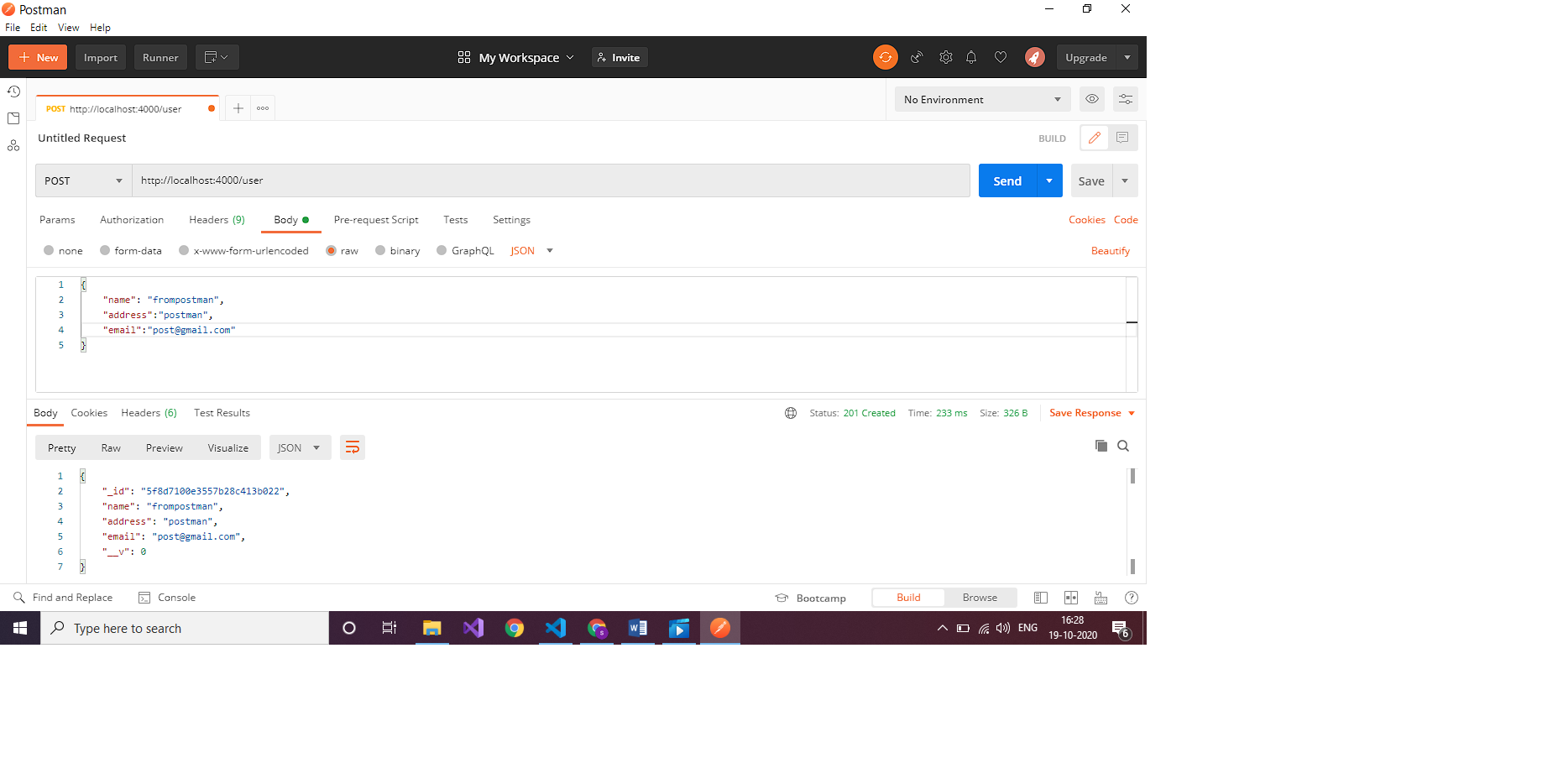
           res.status(201).json(result)

       })

})

app.listen(4000)

1. in POSTMAN type data like below



# DELETE API

//for GET API

const express = require('express');

const app = express();

const mongoose = require('mongoose');

var bodyparder = require('body-parser'); // for post api

var jsonparser = bodyparder.json();

const User = require('./models/getAPI');//model

const { Console } = require('console');

mongoose.connect('mongodb+srv://brijesh:bkrk0403@cluster1.8qcjw.mongodb.net/demo?retryWrites=true&w=majority',

    {

        useNewUrlParser: true,

        useUnifiedTopology: true

    }

);

// for GET API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

//for all record

// app.get('/users', (req,res)=>{

//    User.find().then((data)=>{ //then is promise

//         res.json(data)

//    })

// })

//for specific record

// app.get('/users', (req,res)=>{

//     User.find().select('name').then((data)=>{ //select field name like name ,age

//          res.json(data)

//     })

//  })

// for post API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// app.post('/user', jsonparser,(req,res)=>{ // 2nd parameter is middleware

//     const data = new User({

//         \_id:new mongoose.Types.ObjectId(),

//          name:req.body.name,

//           address:req.body.address,

//           email:req.body.email

//     })

//        data.save().then((result)=>{ //then is promise

//            res.status(201).json(result)

//        })

// })

//for DELETE API

app.delete('/user/:id', (req,res)=>{

    User.deleteOne({\_id:req.params.id}).then((result)=>{

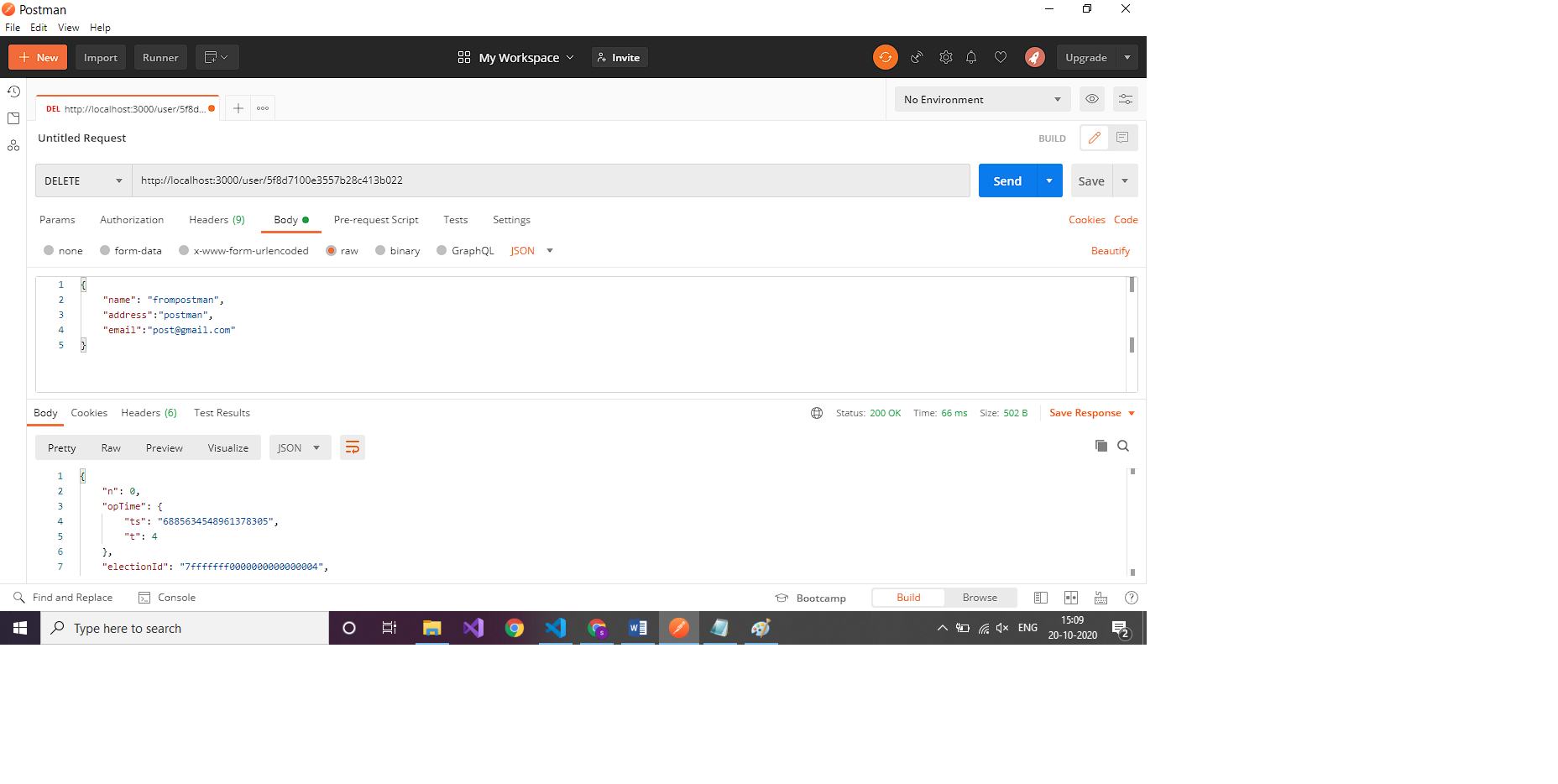
        res.status(200).json(result)

    }).catch((err)=>{Console.warn(err)})

})

app.listen(3000)

in postman



# PUT API

const express = require('express');

const app = express();

const mongoose = require('mongoose');

var bodyparder = require('body-parser'); // for post api

var jsonparser = bodyparder.json();

const User = require('./models/getAPI');//model

const { Console } = require('console');

mongoose.connect('mongodb+srv://brijesh:bkrk0403@cluster1.8qcjw.mongodb.net/demo?retryWrites=true&w=majority',

    {

        useNewUrlParser: true,

        useUnifiedTopology: true

    }

);

// for GET API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

//for all record

// app.get('/users', (req,res)=>{

//    User.find().then((data)=>{ //then is promise

//         res.json(data)

//    })

// })

//for specific record

// app.get('/users', (req,res)=>{

//     User.find().select('name').then((data)=>{ //select field name like name ,age

//          res.json(data)

//     })

//  })

// for post API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// app.post('/user', jsonparser,(req,res)=>{ // 2nd parameter is middleware

//     const data = new User({

//         \_id:new mongoose.Types.ObjectId(),

//          name:req.body.name,

//           address:req.body.address,

//           email:req.body.email

//     })

//        data.save().then((result)=>{ //then is promise

//            res.status(201).json(result)

//        })

// })

//for DELETE API

// app.delete('/user/:id', (req,res)=>{

//     User.deleteOne({\_id:req.params.id}).then((result)=>{

//         res.status(200).json(result)

//     }).catch((err)=>{Console.warn(err)})

// })

app.put('/user/:id', jsonparser , (req,res)=>{

    User.updateOne(

        {\_id:req.params.id},

        { $set: {

            name:req.body.name,

            address: req.body.address,

            email: req.body.email

        }}

    ).then((result)=>{

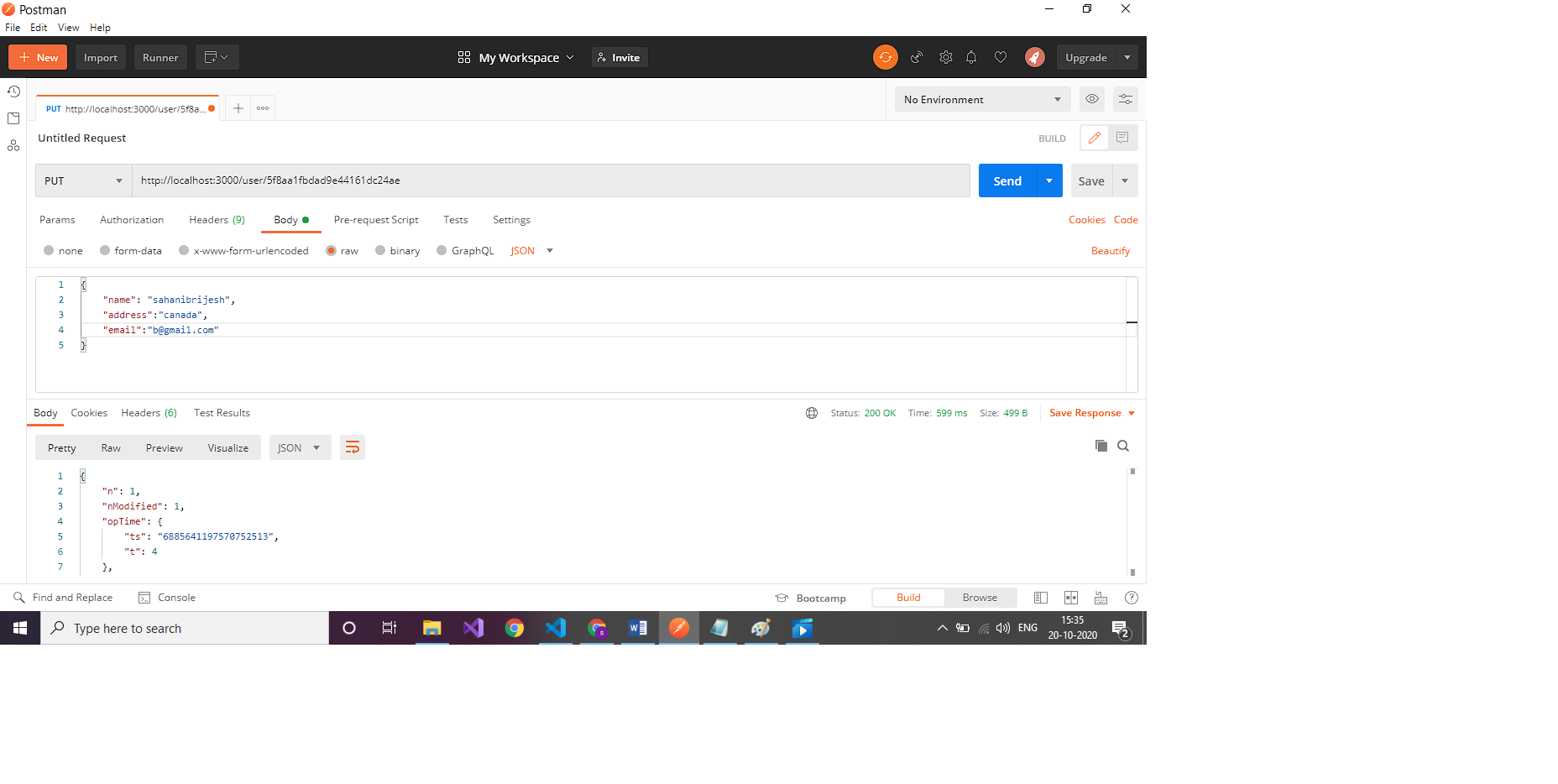
        res.status(200).json(result)

    }).catch((err)=>{Console.warn(err)})

})

app.listen(3000)

in postman



# SEARCH API

const express = require('express');

const app = express();

const mongoose = require('mongoose');

var bodyparder = require('body-parser'); // for post api

var jsonparser = bodyparder.json();

const User = require('./models/getAPI');//model

const { Console } = require('console');

mongoose.connect('mongodb+srv://brijesh:bkrk0403@cluster1.8qcjw.mongodb.net/demo?retryWrites=true&w=majority',

    {

        useNewUrlParser: true,

        useUnifiedTopology: true

    }

);

// for GET API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

//for all record

// app.get('/users', (req,res)=>{

//    User.find().then((data)=>{ //then is promise

//         res.json(data)

//    })

// })

//for specific record

// app.get('/users', (req,res)=>{

//     User.find().select('name').then((data)=>{ //select field name like name ,age

//          res.json(data)

//     })

//  })

// for post API>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// app.post('/user', jsonparser,(req,res)=>{ // 2nd parameter is middleware

//     const data = new User({

//         \_id:new mongoose.Types.ObjectId(),

//          name:req.body.name,

//           address:req.body.address,

//           email:req.body.email

//     })

//        data.save().then((result)=>{ //then is promise

//            res.status(201).json(result)

//        })

// })

//for DELETE API

// app.delete('/user/:id', (req,res)=>{

//     User.deleteOne({\_id:req.params.id}).then((result)=>{

//         res.status(200).json(result)

//     }).catch((err)=>{Console.warn(err)})

// })

//for PUT API

// app.put('/user/:id', jsonparser , (req,res)=>{

//     User.updateOne(

//         {\_id:req.params.id},

//         { $set: {

//             name:req.body.name,

//             address: req.body.address,

//             email: req.body.email

//         }}

//     ).then((result)=>{

//         res.status(200).json(result)

//     }).catch((err)=>{Console.warn(err)})

// })

//for SEARCH API

app.get('/search/:name', (req,res)=>{

        var regex = new RegExp(req.params.name, 'i') //i for remove casesensitive

        User.find({name:regex}).then((result)=>{

            res.status(200).json(result)

       }).catch((err)=>{Console.warn(err)})

     })

app.listen(3000)

in postman

