|  |
| --- |
| **Restfull API : CRUD**  npm init -y  npm install express  npm install mongoose  npm install -g nodemon or sudo npm install -g nodemon |
| Server.js  const express = require('express')  const mongoose = require('mongoose')  const userRouter = require('./routes/users')  const url = "mongodb+srv://gauravraj:Mongo%40raj11@cluster0.cddlpyx.mongodb.net/crud?retryWrites=true&w=majority&appName=Cluster0"  //created server  const app = express()  //mongodb connections  mongoose.connect(url)  const con = mongoose.connection  con.on('open', () => {  console.log('connected...')  })  //middlewares  app.use(express.json())  //api endpoints  app.use('/users',userRouter) // localhost:9000/users  //listening port  app.listen(9000, () => {  console.log('Server started')  }) |
| .routes/users.js  const express = require('express')  const router = express.Router()  const User = require('../models/user')  router.get('/', async(req,res) => {  try{  const users = await User.find()  res.json(users)  }catch(err){  res.send('Error ' + err)  }  })  router.get('/:id', async(req,res) => {  try{  const user = await User.findById(req.params.id)  res.json(user)  }catch(err){  res.send('Error ' + err)  }  })  router.post('/', async(req,res) => {  const user = new User({  name: req.body.name,  tech: req.body.tech,  sub: req.body.sub  })  try{  const a1 = await user.save()  res.json(a1)  }catch(err){  res.send('Error')  }  })  router.patch('/:id',async(req,res)=> {  try{  const user = await User.findById(req.params.id)  user.sub = req.body.sub  const a1 = await user.save()  res.json(a1)  }catch(err){  res.send('Error')  }  })  // Delete a user  router.delete('/:id', getUser, async (req, res) => {  try {  // await res.user.remove();  await res.user.deleteOne();  res.json({ message: 'Deleted User' });  } catch (err) {  res.status(500).json({ message: err.message });  }  });  async function getUser(req, res, next) {  let user;  try {  user = await User.findById(req.params.id);  if (user == null) {  return res.status(404).json({ message: 'Cannot find user' });  }  } catch (err) {  return res.status(500).json({ message: err.message });  }  res.user = user;  next();  }  module.exports = router |
| ./models/user  const mongoose = require('mongoose')  const userSchema = new mongoose.Schema({  name: {  type: String,  required: true  },  tech: {  type: String,  required: true  },  sub: {  type: Boolean,  required: true,  default: false  }  })  module.exports = mongoose.model('User',userSchema) |