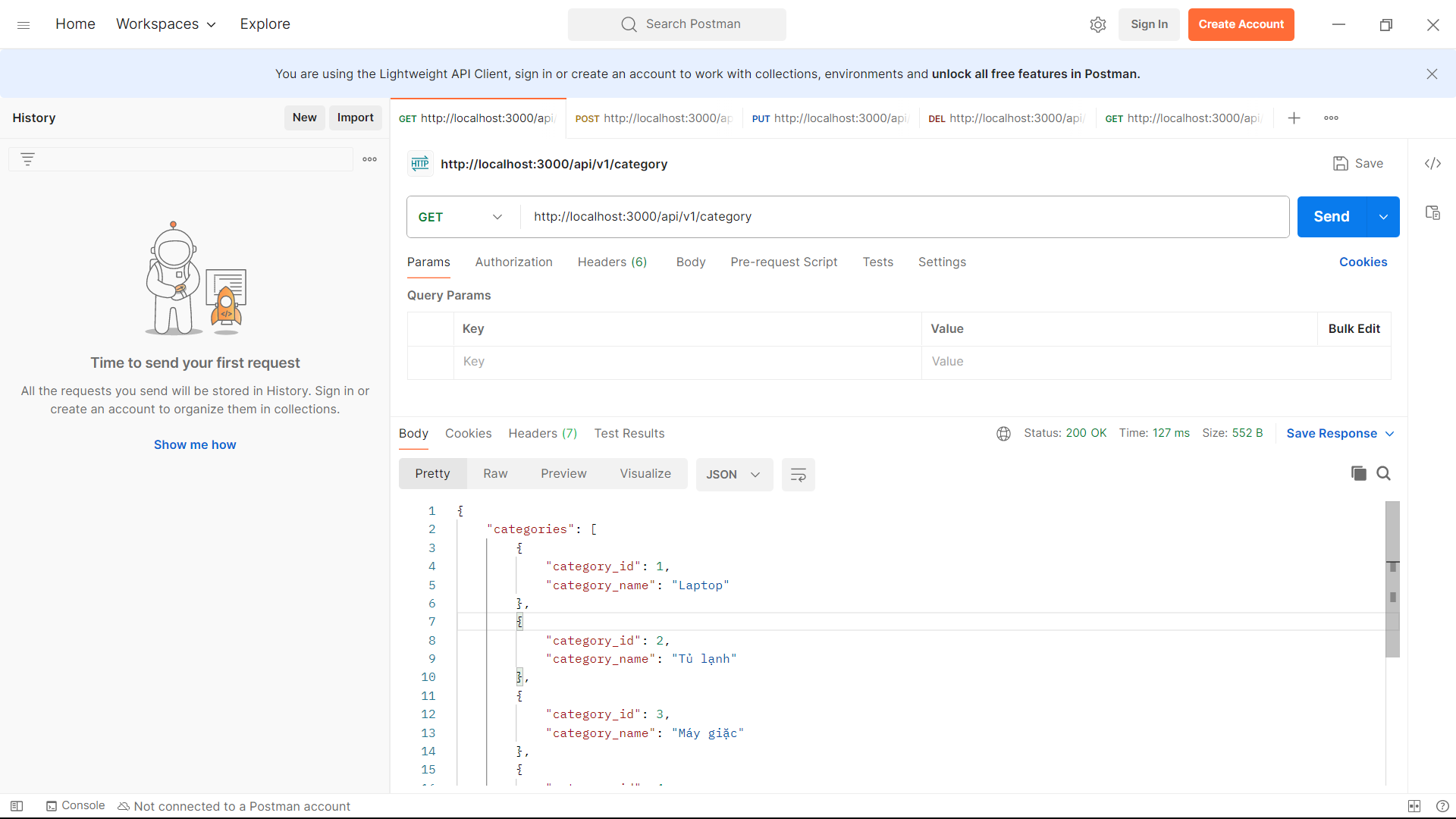
Tải phần mềm postman <https://www.postman.com/downloads/>



Bước 1: tạo thư mục project

Bước 2: sử dụng cmd dưới để khởi tạo project nodejs

npm init --yes

npm install

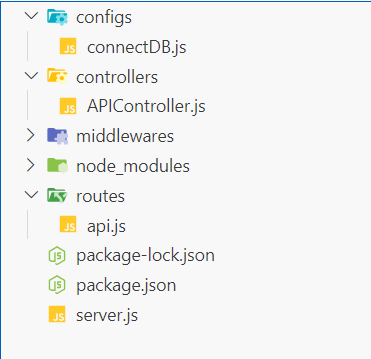
cài đặt framwork expressjs & mysql driver

npm install express --save

npm install mysql2 --save

npm install body-parser --save

Bước 3: tạo cấu trúc thư mục và file



Bước 4: cấu hình file connectDB.js kết nối tới Database

const mysql = require('mysql2/promise');//thêm thư viện để sử dụng Promise và async/await trong việc thao tác với cơ sở dữ liệu MySQL

const pool = mysql.createPool({

    host: 'localhost',

    user: 'root',

    database: 'demo',

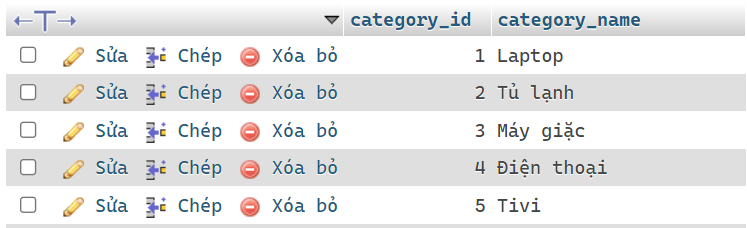
    password: ''

})

module.exports = pool;//trả về giá trị để có thể sử dụng vào các file khác

Bước 5: cấu hình các hàm CRUD cho các bảng trong database trong file APIControler.js

Ví dụ bảng tbl\_category(category\_id, category\_name)



const pool = require('../configs/connectDB');

const express = require('express');

const app = express();

app.use(express.json());

const getAllCategories = async (req, res) => {

    try {

        const [rows, fields] = await pool.execute('SELECT \* FROM tbl\_category');

        res.status(200).json({ categories: rows });

    } catch (error) {

        console.error('Error fetching categories:', error);

        res.status(500).json({ error: 'Internal server error' });

    }

};

const createCategory = async (req, res) => {

    const { category\_name } = req.body || {};

    if (!category\_name || typeof category\_name !== 'string') {

        return res.status(400).json({ message: 'Missing required params: category\_name' });

    }

    try {

        const [result] = await pool.execute('INSERT INTO tbl\_category (category\_name) VALUES (?)', [category\_name]);

        return res.status(201).json({ message: 'Category created successfully' });

    } catch (error) {

        console.error('Error creating category:', error);

        return res.status(500).json({ message: 'Internal server error' });

    }

};

const updateCategory = async (req, res) => {

    const { category\_id, category\_name } = req.body || {};

    // Check for missing and invalid parameters

    if (!category\_id || typeof category\_id !== 'number' || !category\_name || typeof category\_name !== 'string') {

        return res.status(400).json({ message: 'Missing or invalid required params' });

    }

    try {

        const [result] = await pool.execute('SELECT \* FROM tbl\_category WHERE category\_id = ?', [category\_id]);

        if (result.length === 0) {

            return res.status(404).json({ message: 'Category not found' });

        }

        await pool.execute('UPDATE tbl\_category SET category\_name = ? WHERE category\_id = ?', [category\_name, category\_id]);

        return res.status(200).json({ message: 'Category updated successfully' });

    } catch (error) {

        console.error('Error updating category:', error);

        return res.status(500).json({ message: 'Internal server error' });

    }

};

const deleteCategory = async (req, res) => {

    const { category\_id } = req.body || {};

    if (!category\_id) {

        return res.status(400).json({ message: 'Missing or invalid required param: category\_id (must be a number)' });

    }

    try {

        const [result] = await pool.execute('DELETE FROM tbl\_category WHERE category\_id = ?', [category\_id]);

        return res.status(200).json({ message: 'Category deleted successfully' });

    } catch (error) {

        console.error('Error deleting category:', error);

        return res.status(500).json({ message: 'Internal server error' });

    }

};

module.exports = { getAllCategories, createCategory, updateCategory, deleteCategory };

Bước 6: cấu hình các router trong file api.js trong thư mục routes

const express = require("express");

const APIController = require('../controllers/APIController');

let router = express.Router();

const initAPIRoute = (app) => {

    router.get('/category', APIController.getAllCategories); // method GET -> READ data

    router.post('/create-category', APIController.createCategory); // method POST -> CREATE data

    router.put('/update-category', APIController.updateCategory); //method PUT -> UPDATE data

    router.delete('/delete-category/:id', APIController.deleteCategory); //method DELETE -> DELETE data

    router.get('/sanpham', APIController.getAllSanpham);

    return app.use('/api/v1/', router)

}

module.exports = initAPIRoute; //trả về giá trị để tiếp tục sử dụng trong các file khác

Bước 7: cấu hình file server.js để chạy project

const express = require('express');

const initAPIRoute = require('./routes/api');

const app = express();

app.use(express.json());

const port = 3000; //xác định port để thực hiện project

initAPIRoute(app);

app.listen(port, () => {

    console.log(`Server is running at http://localhost:${port}`);

});

//hiện thông báo cho biết máy chủ đang chạy ở địa chỉ <http://localhost:3000>

Bước 8: chạy server bằng câu lệnh npm start