

## LAB 04-RESTFUL API TRONG NODEJS-MongoDB

**RESTFUL API** trong NodeJS là cách triển khai web service nhằm public các tài nguyên của mình phục vụ các request từ xa.

### RESTful API trong NodeJS

REST là gì? REST – viết đầy đủ là REpresentational State Transfer – là một chuẩn trong lập trình. Nó định nghĩa các quy tắc để tạo ra các web service nhằm cung cấp các chức năng truy xuất, thêm mới, cập nhật tài nguyên từ xa thông qua http. Restful API diễn dịch lại dữ liệu giữa các hệ thống ứng dụng và máy tính, tạo nên khả năng truyền tải dữ liệu đơn giản mà hiệu quả, phục vụ trao đổi dữ liệu giữa các nền tảng khác nhau và xa nhau.

Restful, tạo nên khả năng truy cập tài nguyên cho client thông qua các URI. Dữ liệu trả về sẽ trong dạng json, xml, html nhưng chủ yếu là json. Restful sử dụng HTTP như cỗ xe chuyên chở dữ liệu giữa server và client qua các method GET, HEAD, POST, PUT, PATCH, DELETE...

### CÁC DẠNG DỮ LIỆU TRẢ VỀ TRONG RESTFUL API

Khi dùng restful, dữ liệu truyền giữa client và server là dạng json, đây là dạng dữ liệu có cấu trúc, rất ngắn gọn. Ví dụ

```
{ "thongbao": "Đã thêm thành công" }  
hoặc  
{"id": "1", "tensp": "abc", "gia": "250000" }  
hoặc  
[  
  {"id": "1", "tenSach": "abc", "moTa": "...", "urlHinh": "..."}  
  ,  
  {"id": "2", "tenSach": "xyz", "moTa": "...", "urlHinh": "..."}  
  ,  
  {"id": "3", "tenSach": "mnp", "moTa": "...", "urlHinh": "..."}  
]
```

## CÁC QUY ĐỊNH HTTP METHOD TRONG RESTFUL

HTTP method	Mô tả
GET	Dùng để lấy một tài nguyên hoặc danh sách các tài nguyên từ server (đã representation lại theo dạng json, xml...)
POST	Gửi đến server tài nguyên để xử lý (thường dùng cho thêm mới record)
PUT	Cập nhật trạng thái mới cho tài nguyên trên server (thường dùng cho update record)
DELETE	Xóa tài nguyên chỉ định trên server

### Các status code http

Khi request đến một tài nguyên, các status code sẽ trả về client :

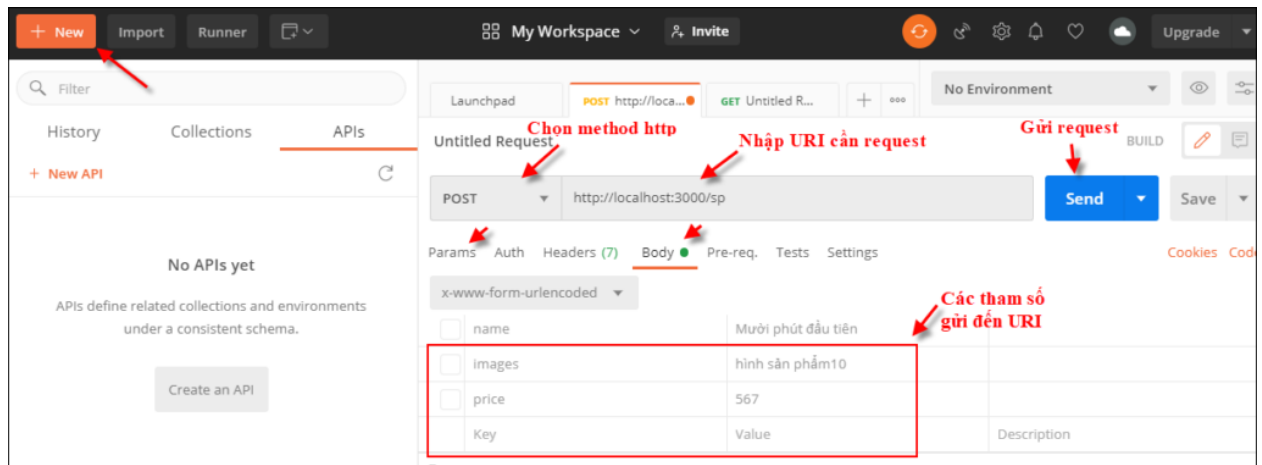
- 200 OK – Tra về trong các phương thức GET, PUT, PATCH, DELETE.
- 201 Created – Trả về khi tạo xong tài nguyên
- 204 No Content – Trả về khi xóa xong 1 tài nguyên
- 304 Not Modified – Tài nguyên không có thay đổi, client có thể dùng cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request không quyền truy cập.
- 403 Forbidden – Request bị bẻ gãy, từ chối truy cập.
- 404 Not Found – Không tìm thấy tài nguyên trong từ URI
- 405 Method Not Allowed – Phương thức không được phép
- 410 Gone – Tài nguyên không tồn tại
- 415 Unsupported Media Type – Không hỗ trợ kiểu tài nguyên
- 422 Unprocessable Entity – Dữ liệu không được xử lý
- 429 Too Many Requests – Có quá nhiều request

## SỬ DỤNG POSTMAN ĐỂ TEST RESTFUL API

Postman là một chương trình đóng vai 1 client để tạo request đến server. Postman như một trình duyệt dành cho web dev. Nó hỗ trợ đầy đủ các method http như post, get, put, delete....và các tham số trong http request.

Địa chỉ để download là <https://www.postman.com/downloads/>

Cài đặt postman: Chạy chương trình, nhấp nút next next là xong



**Tạo RESTful API trong NodeJS**, các em cần chuẩn bị database, tạo folder project trước khi thực hiện.

### Chuẩn bị database trong mongodb: qlsach

Tạo table tên **sach** như sau: id(tự động), tenSach, moTa, urlHinh, capNhat, gia, idLoai, anHien

Nhập dữ liệu (Xem lại bài Lab03) với các dữ liệu như bảng dưới đây

## Insert Document

tenSach	moTa	urlHinh	capNhat	gia	idLoai	anHien
Sống Chạm Đợi Nắng Lên	Lúc con trai còn nhỏ, tôi thường có những trò chơi...	images/song-cham-doi-nang-len.png	2022-01-08 14:10:27	89000	3	1
Sống Xanh Như Những Lá Trà	Trong cuốn sách thú vị này, bạn sẽ được khám phá k...	images/song-xanh-nhu-nhung-la-tra.png	2022-01-08 14:10:04	109000	3	1
Sống Như Lần Đầu, Yêu Như Lần Cuối	Cuốn sách là những câu chuyện đời sống hằng ngày, ...	images/song-nhu-lan-dau-tien-yeu-nhu-lan-cu...	2022-01-08 14:13:40	99000	3	1
Những Ngày Đầy Nắng	"Bạn thân mến! Tôi chưa bao giờ nghĩ mình sẽ viết ...	images/nhung-ngay-day-nang.png	2022-01-08 14:15:44	129000	3	1
Sức Mạnh Của Sự Từ Tế	Bạn có nhớ nụ cười của một người lạ khi bạn giữ cừ...	images/suc-manh-cua-su-tu-te.png	2022-01-08 14:18:44	68000	3	1
Đề Có Một Tương Lai	Trong xã hội có rất nhiều mối nguy. Nếu chúng ta đ...	images/de-co-mot-tuong-lai.png	2022-01-08 14:33:14	155000	3	1
Cân Bằng Cảm Xúc, Cả Lúc Bão Giông	Một ngày, chúng ta có khoảng 16 tiếng tiếp xúc với...	images/can-bang-cam-xuc-ca-luc-bao-giong.jpg	2022-01-08 14:34:50	96000	3	1
Hiếu Về Trái Tim	Xuất bản lần đầu tiên vào năm 2011, Hiếu Về Trái T...	images/hieu-ve-trai-tim.jpg	2022-01-08 15:01:03	138000	3	1

Cài module express và express-generator

Nếu máy của các em đã cài rồi thì thôi bỏ qua, còn chưa thì

1. Mở command line
2. Gõ lệnh **npm install -g express**
3. Tiếp theo gõ lệnh **npm -g install express-generator**

## Tạo project

### 1. Tạo project NodeJS và cài module

Chạy lệnh tạo project:

```
express -ejs Node_Mongodb
```

Cài đặt các module vào project: Mở command line rồi chuyển vào folder project gõ lệnh:

```
npm install  
npm install mongodb
```

### 2. Nhúng hàm MongoClient trong module mongodb vào

#### Tạo model kết nối db

1. Tạo folder models trong project
2. Tại models tạo file database.js để kết nối database “qlsach”

```
const { MongoClient } = require("mongodb");
```

```
const uri = "mongodb://localhost:27017";  
const client = new MongoClient(uri);  
await client.connect();  
console.log('Kết nối thành công đến server');  
const db = client.db('qlsach');
```

#### Tạo route

1. Mở file app.js và định nghĩa route **sach** dẫn vào controller **routes/sach**

```
var sachRouter = require('./routes/sach');  
app.use('/sach', sachRouter);
```

## 2. Tạo file routes/sach.js và code

```
var express = require('express');
var router = express.Router();
var db = require('../models/database');
router.get('/', function(req, res) {
    //chức năng trả về danh sách các record
    //lấy ra các record trong table
    //trả về các record dạng json
});
router.post('/', function(req, res) {
    //chức năng thêm mới record vào table
    //tiếp nhận dữ liệu gửi trong body request
    //thực hiện chèn record mới vào table
    //trả về thông báo đã chèn dạng json
});
router.get('/:id', function(req, res) {
    //chức năng trả về chi tiết 1 record
    //tiếp nhận id của reord trong url
    //lấy ra record theo id từ table
    //trả về chi tiết record dạng json
});
router.put('/:id', function(req, res) {
    //chức năng cập nhật record trong table
    //tiếp nhận dữ liệu gửi trong body request
    //thực hiện cập nhật record vào table
    //trả về thông báo json đã cập nhật
});
router.delete('/:id', function(req, res) {
    //chức năng xóa 1 record trong table
    //tiếp nhận id trong url
    //thực hiện xóa record
    //trả về thông báo json đã xóa
});
module.exports = router;
```

## ❖ THỰC HIỆN CHỨC NĂNG DANH SÁCH RECORD

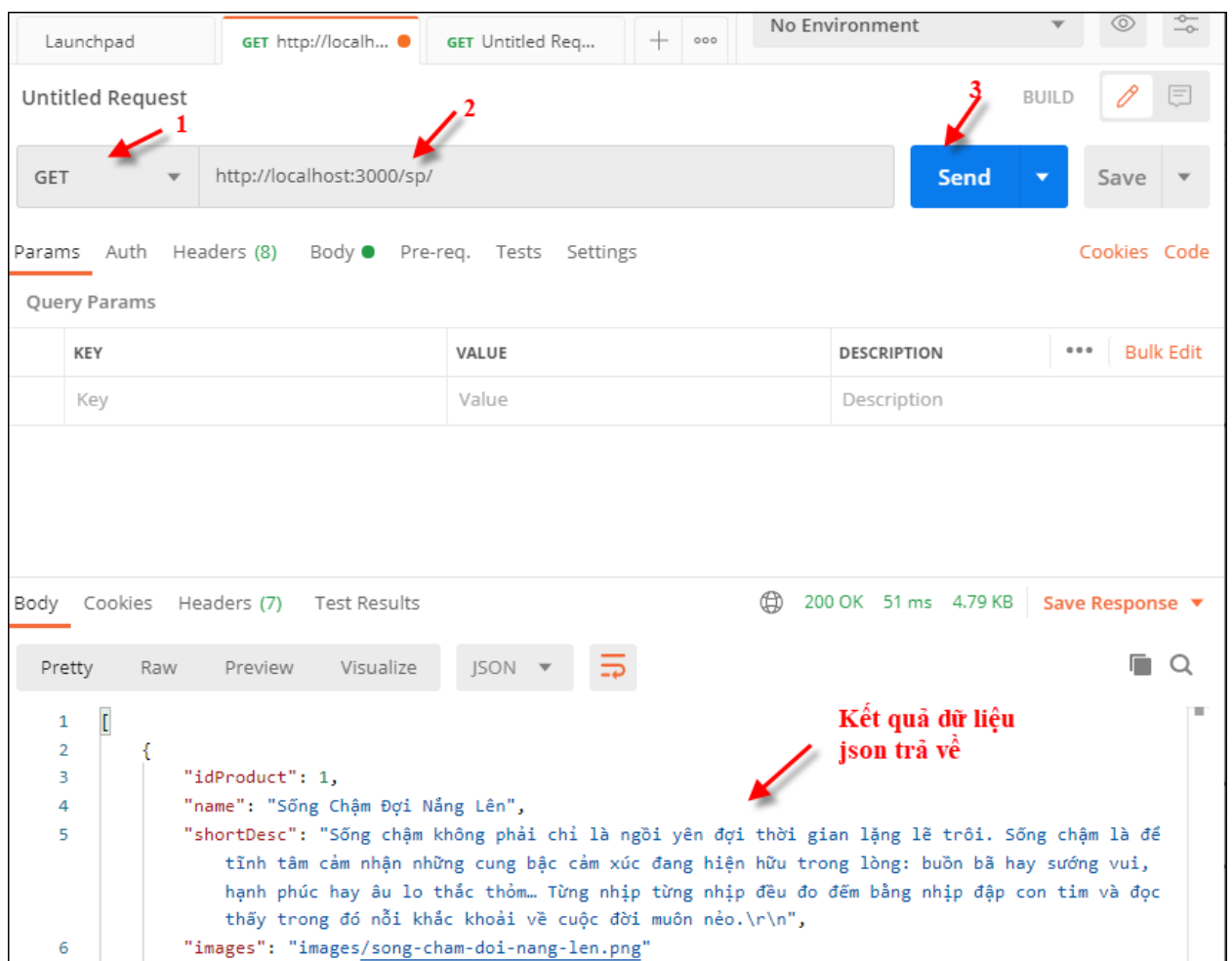
1. Trong `routes/sach.js`, code lại chức năng trả về danh sách các record để được như sau

```
router.get('/', function(req, res) {  
  //chức năng trả về danh sách các record  
  //lấy ra các record trong table  
  //trả về các record dạng json  
});
```

Xem lại lab03 để code cho phần lấy danh sách

2. Test: Mở Postman để test

Mở chương trình **postman**, chọn method GET và nhập url rồi nhấn Send như hình



## ❖ THỰC HIỆN LẤY CHI TIẾT RECORD

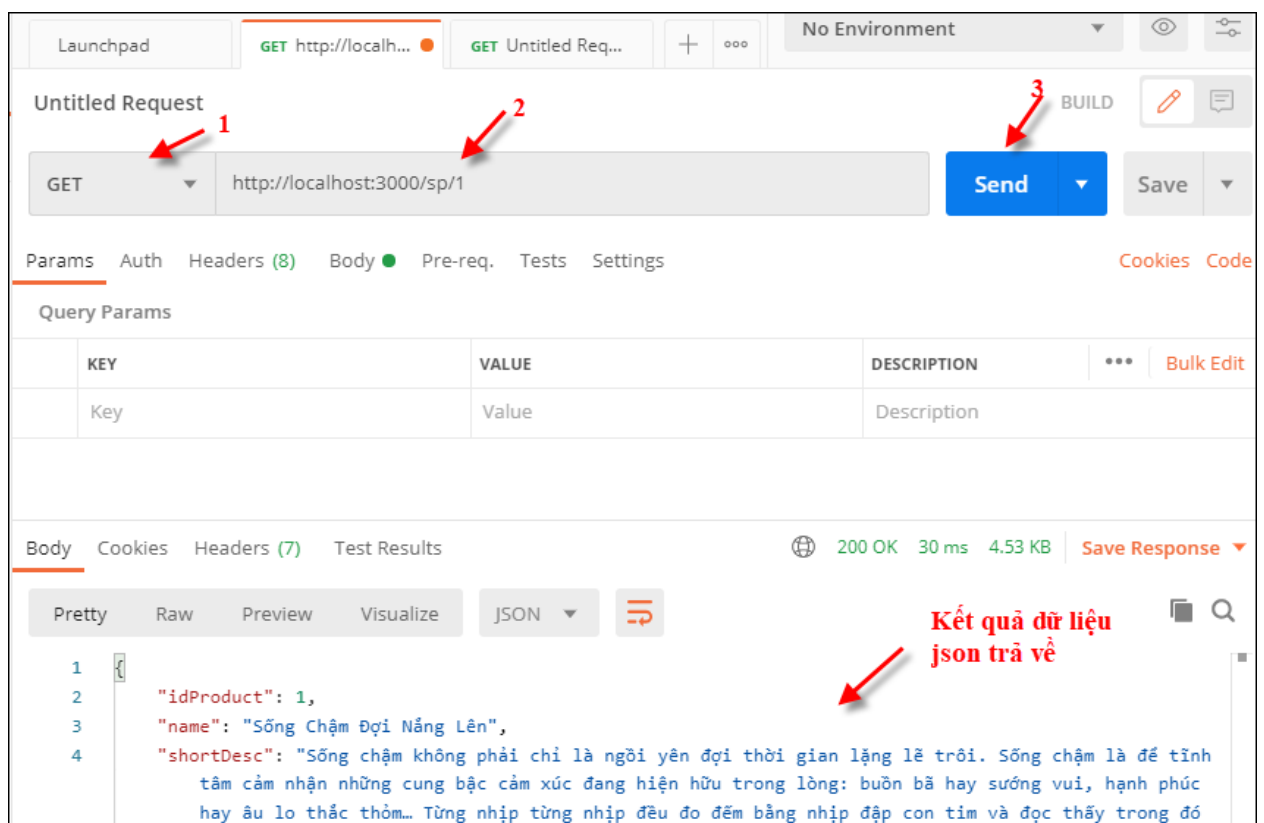
1. Trong routes/sach.js, code lại **chức năng** trả về chi tiết 1 record để được như sau

```
router.get('/:id', function(req, res) {  
  //chức năng trả về chi tiết 1 record  
  //tiếp nhận id của reord trong url  
  //lấy ra record theo id từ table  
  //trả về chi tiết record dạng json  
});
```

Xem lại lab03 để code cho phần lấy chi tiết sách theo id

2. Test: Mở Postman để test

Trong tool **postman**, chọn method GET và nhập url rồi nhấp Send như hình





## ❖ THỰC HIỆN CHỨC NĂNG THÊM RECORD

1. Trong `routes/sach.js`, code lại **chức năng thêm mới record** vào **table** để được như sau:

```
router.post('/', function(req, res) {  
  //chức năng thêm mới record vào table  
  //tiếp nhận dữ liệu gửi trong body request  
  //thực hiện chèn record mới vào table  
  //trả về thông báo đã chèn dạng json  
});
```

Xem lại lab03 để code cho phần thêm

2. Test: Mở Postman để test

Trong tool **postman**, chọn method POST và nhập url rồi nhấn Send như hình

The screenshot shows the Postman interface for a POST request. The request is configured with the method **POST** and the URL `http://localhost:3000/sp/`. The body is set to `x-www-form-urlencoded` and contains three fields: **name** (Mười phút đầu tiên), **images** (hình sản phẩm10), and **price** (567). The response is shown in JSON format, indicating a successful status of 200 OK with a message: `"thongbao": "Đã chèn xong một sản phẩm mới"`.

Annotations in the image:

- 1: Points to the **POST** method dropdown.
- 2: Points to the URL `http://localhost:3000/sp/`.
- 3: Points to the **Send** button.
- 4: Points to the **Body** tab.
- 5: Points to the **JSON** response.

**Kết quả dữ liệu json trả về**



## ❖ THỰC HIỆN CHỨC NĂNG CẬP NHẬT RECORD

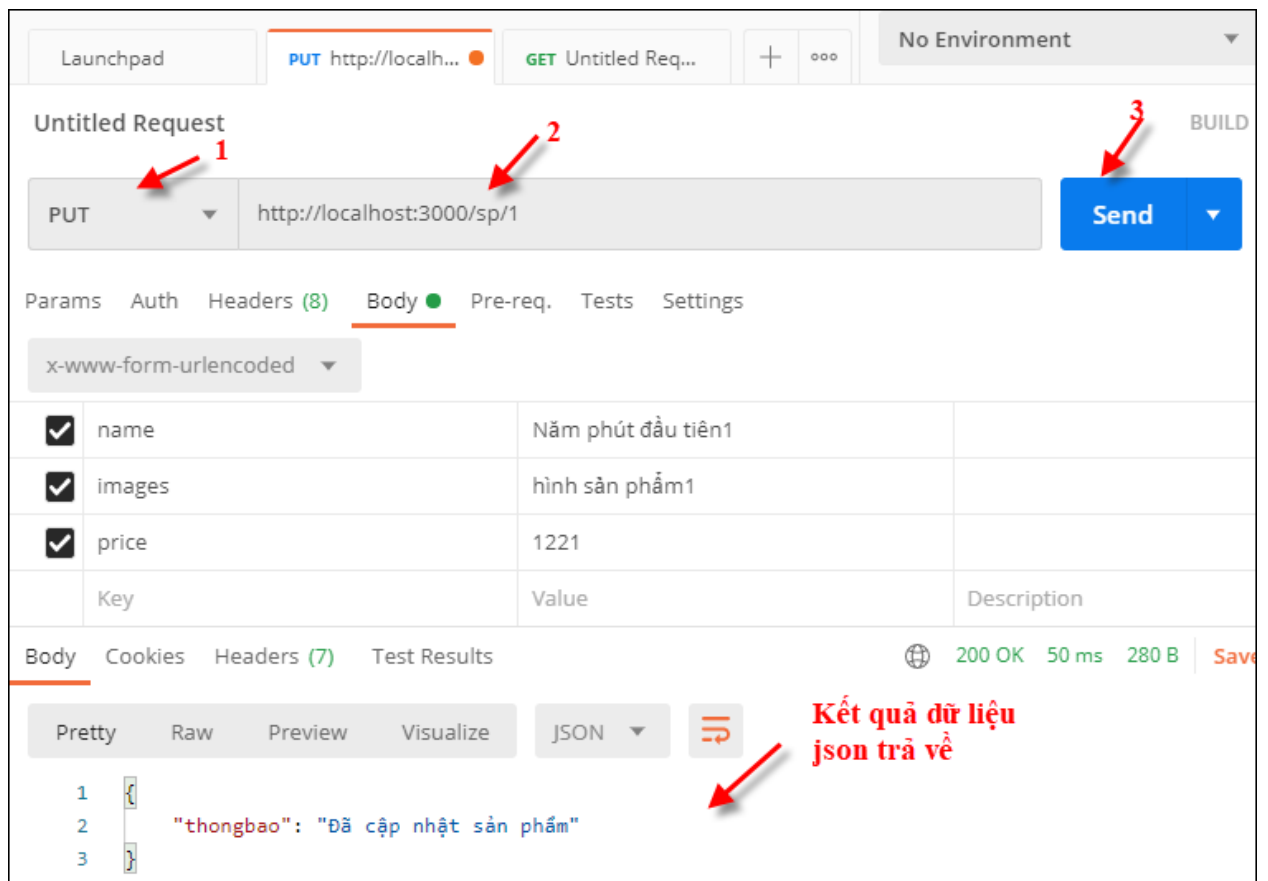
1. Trong `routes/sach.js`, code lại chức năng cập nhật record trong table để được như sau:

```
router.put('/:id', function(req, res) {
  //chức năng cập nhật record trong table
  //tiếp nhận dữ liệu gửi trong body request
  //thực hiện cập nhật record vào table
  //trả về thông báo json đã cập nhật
});
```

Xem lại lab03 để code cho phần cập nhật

2. Test: Mở Postman để test

Trong tool **postman**, chọn method **PUT** và nhập url rồi nhấn Send như hình



## ❖ THỰC HIỆN CHỨC NĂNG XÓA RECORD

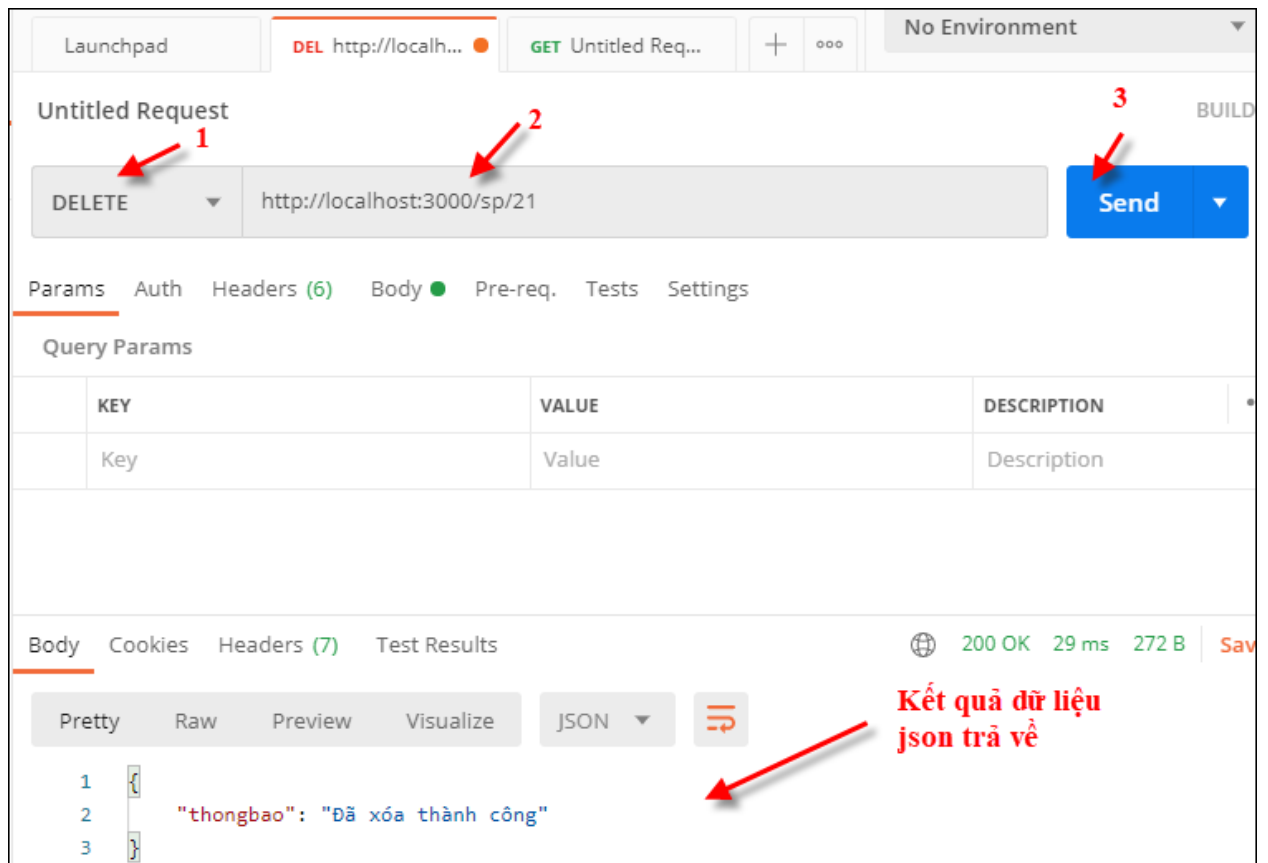
1. Trong `routes/sach.js`, code lại **chức năng xóa record trong table** để được như sau:

```
router.delete('/:id', function(req, res) {  
  //chức năng xóa 1 record trong table  
  //tiếp nhận id trong url  
  //thực hiện xóa record  
  //trả về thông báo json đã xóa  
});
```

Xem lại lab03 để code cho phần xóa

2. Test: Mở Postman để test

Trong tool **postman**, chọn method **DELETE** và nhập url rồi nhấn Send như hình



HẾT