

VNUHCM, University of Science



Lab 08:

ExpressJS

Phát triển ứng dụng Web



1 Mục tiêu

Hướng dẫn sinh viên xây dựng Restful API với ExpressJS

2 Quy định bài thực hành và định dạng nộp bài

- Thời gian thực hiện: **4 tiết**
- Sinh viên sẽ tạo website với cấu trúc thư mục theo định dạng: **<MSSV>_lab08**
- Thực hiện lại tất cả các bước trong phần **hướng dẫn thực hành** (bên dưới).
- Sau đó nén toàn bộ folder **<MSSV>_lab08** lại file zip hoặc rar và nộp lên hệ thống Moodle.
- Công cụ sử dụng để lập trình là **Visual Code**
- Yêu cầu:
 - o Phần resource (tài nguyên): được cung cấp trong thư mục images
 - o Các trường hợp làm bài không đúng qui định sẽ không được chấp nhận.

3 Hướng dẫn thực hành

Bước 01

Thực hiện tạo project **NodeJS** với lệnh **\$ npm init** như **Lab07**

Bước 02

Open Workspace trong Visual Code với thư mục vừa **<MSSV>_lab08** tạo

Thực hiện install framework ExpressJS với tập lệnh Terminal trong cửa sổ Terminal của Visual Code như sau:

ExpressJS

```
🍏 > ~/Development/0123456_lab08
> npm install --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 0123456_lab08@1.0.0 No description
npm WARN 0123456_lab08@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 3.672s
found 0 vulnerabilities
```

🔧 Bước 03

File *package.json* sẽ được update thông tin của project với dependencies package như sau:

```
📄 package.json > ...
1  {
2    "name": "0123456_lab08",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    > Debug
7    "scripts": {
8      "start": "node Demo01.js"
9    },
10   "author": "NDHuy",
11   "license": "ISC",
12   "dependencies": {
13     "express": "^4.17.1"
14   }
}
```

Bước 04

Tạo file **Demo01.js** có nội dung như sau:

```
JS Demo01.js > ...
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.listen(port, () => {
6    console.log(`Example app listening at http://localhost:${port}`)
7  })
8
9  app.get('/', (req, res) => {
10    res.send('GET method for READ data')
11  })
12
13 app.post('/', (req, res) => {
14    res.send('POST method for CREATE data')
15  })
16
17 app.put('/', (req, res) => {
18    res.send('PUT method for UPDATE data')
19  })
20
21 app.delete('/', (req, res) => {
22    res.send('DELETE method for DELETE data')
23  })
```

Từ cửa sổ Terminal của Visual code, thực hiện lệnh **\$ node Demo01.js** để deploy file **Demo01.js**

ExpressJS

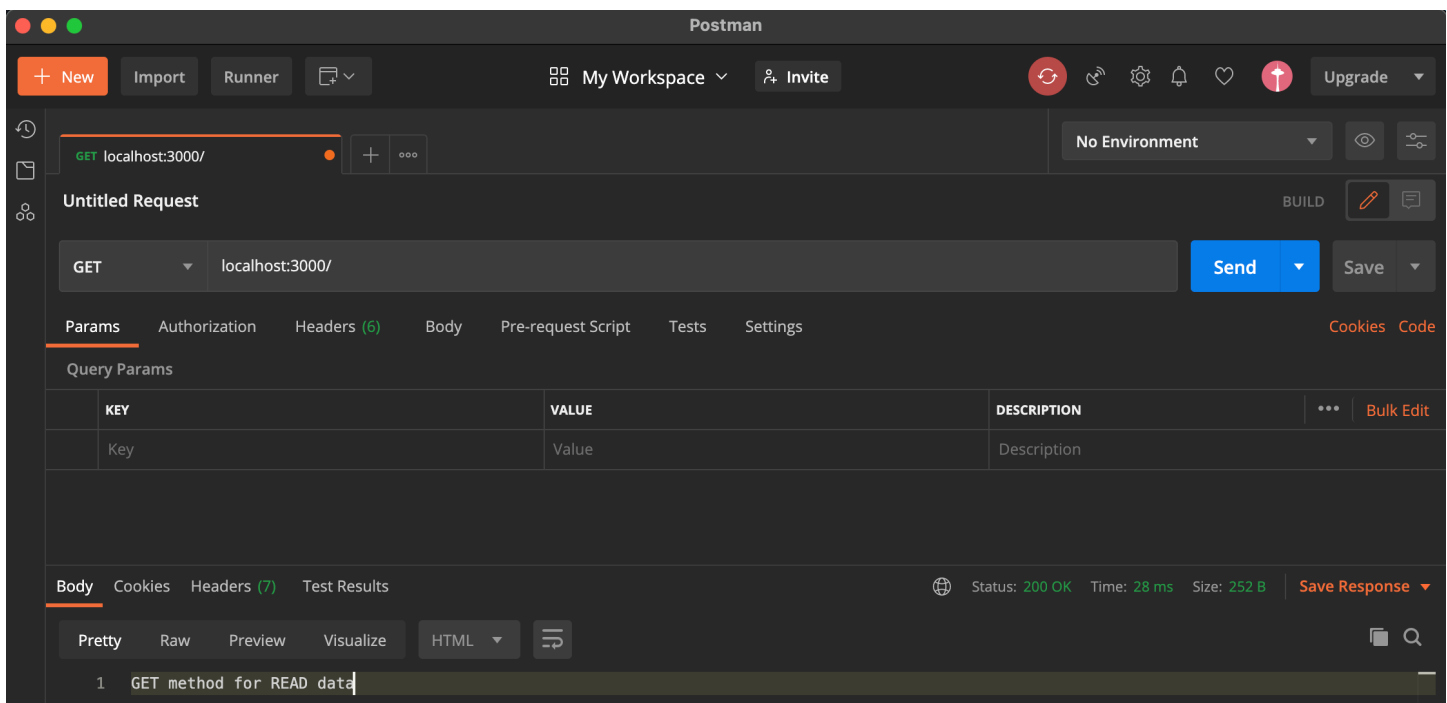
Bước 05

Cài đặt ứng dụng PostMan từ link <https://www.postman.com/> để thực hiện test các API vừa được tạo ra với ExpressJS

Bước 06

Sử dụng PostMan để test các API theo các bước sau:

- Test giao thức GET để lấy dữ liệu từ API



ExpressJS



Page: 5

- Test giao thức POST

The screenshot shows the Postman application window. At the top, there's a toolbar with buttons for '+ New', 'Import', 'Runner', and 'My Workspace'. Below this, the main interface is divided into several sections. The top section shows the request URL 'localhost:3000/' and the method 'POST'. The middle section is titled 'Untitled Request' and contains tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The bottom section is titled 'Body' and contains tabs for 'Cookies', 'Headers (7)', and 'Test Results'. The 'Body' tab is active, showing a text input field with the value 'POST method for CREATE data'.

KEY	VALUE	DESCRIPTION
Key	Value	Description

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 6 ms Size: 255 B Save Response

ExpressJS



Page: 6

- Test giao thức PUT

The screenshot shows the Postman application interface. At the top, there's a toolbar with buttons for '+ New', 'Import', 'Runner', and 'My Workspace'. Below this, the main area is divided into several sections. The top section shows the request URL 'localhost:3000/' and the method 'PUT'. The middle section has tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The bottom section shows the response status '200 OK', time '15 ms', and size '254 B'. The response body is displayed in the bottom-most section, showing the text 'PUT method for UPDATE data'.

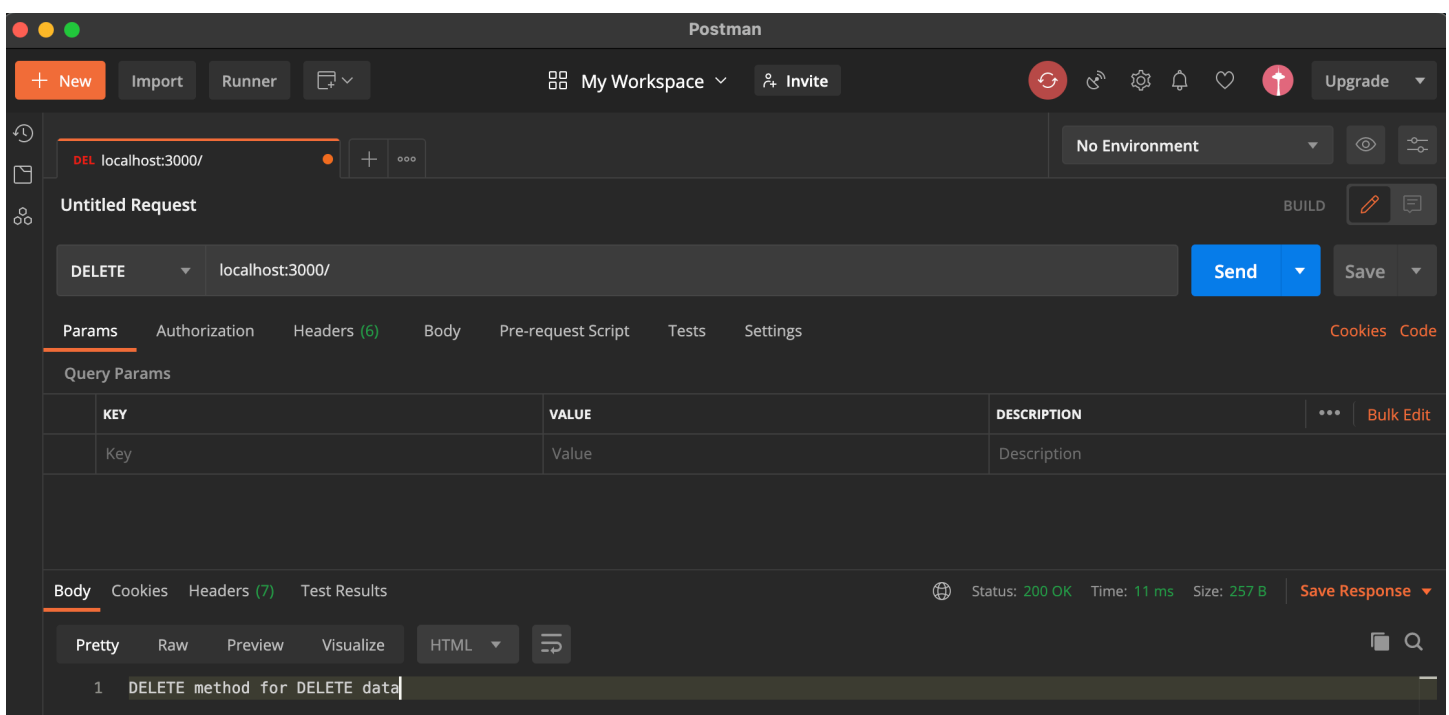
KEY	VALUE	DESCRIPTION
Key	Value	Description

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 15 ms Size: 254 B

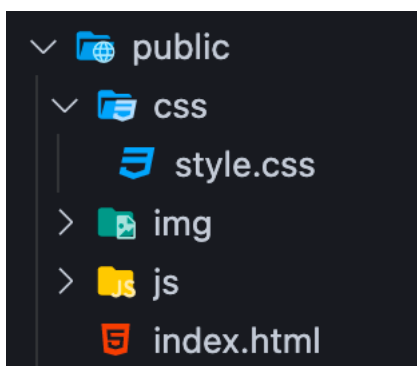
PUT method for UPDATE data

- Test giao thức DELETE



Bước 07

Ở folder SITE_ROOT của project, tạo cấu trúc file và folder như sau:



Trong đó:

- File **style.css**: sẽ chứa mã CSS sẽ được load tự động về Client mỗi khi truy xuất file index.html
- File **index.html**: sẽ chứa mã nguồn HTML tĩnh của Website Bước 08

Thêm nội dung cho file index.html như sau:

```
public > index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <link rel="stylesheet" type="text/css" href="./css/style.css">
8  </head>
9  <body>
10     <h1>Welcome to My Website</h1>
11     <div>This is the first pagraph.</div>
12 </body>
13 </html>
```

Bước 09

Thêm nội dung cho file style.css như sau:

```
public > css >  style.css > ...  
1  h1 { color:  #cc3366; }  
2  
3  div {  
4      color:  #3366cc;  
5      font-size: 13pt;  
6      font-style: italic;  
7  }
```

Bước 10

Ở folder SITE_ROOT sẽ tạo file Demo02.js có nội dung như sau, để truy xuất dữ liệu của index.html khi user truy cập vào url: <http://localhost:3000/>

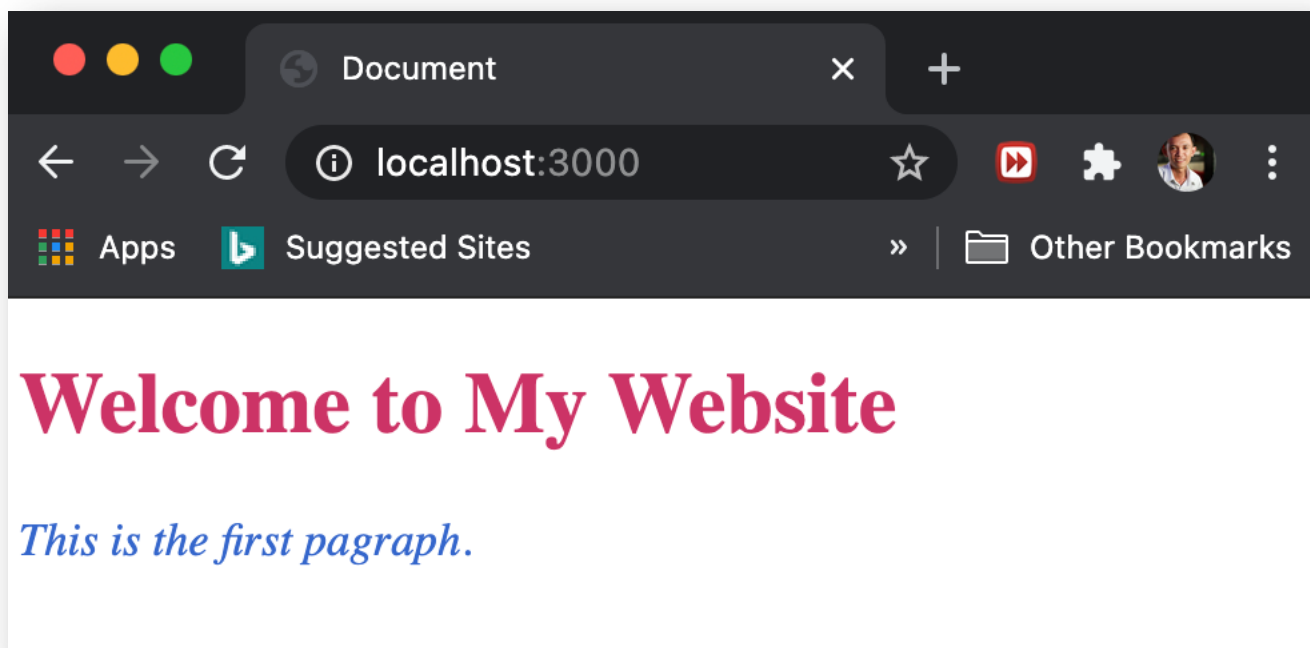
ExpressJS

```
JS Demo02.js > ...
1  const express = require('express')
2  const fs = require('fs')
3
4  const app = express()
5  const port = 3000
6
7  // set thư mục public thành static để cho phép truy cập tự do qua url
8  app.use(express.static('public'))
9
10 app.listen(port, () => {
11   console.log(`Example app listening at http://localhost:${port}`)
12 })
13
14 app.get('/', (req, res) => {
15   fs.readFile('/public/index.html', (error, data) => {
16     if (error)
17       throw error
18     else
19       {
20         res.header(200, {'Content-Type' : 'text/html'})
21         res.send(data)
22       }
23   })
24 })
```

Ở terminal của Visual Code, dùng lệnh \$ node Demo02.js để deploy

Bước 11

Sử dụng Browser (Google Chrome) để truy cập vào link: <http://localhost:3000> sẽ được hiển thị kết quả như sau:



Bước 12

Xây dựng nội dung cho file ***data.json*** như sau:

```
{...} data.json > ...
1  ✓ [
2  ✓   {
3      "id": 1,
4      "name": "iPhone 12"
5  },
6  ✓   {
7      "id": 2,
8      "name": "Galaxy Node 20 Ultra"
9  },
10  ✓   {
11      "id": 3,
12      "name": "MacBook Pro 2020"
13  },
14  ✓   {
15      "id": 4,
16      "name": "MacBook Air 2020 with M1"
17  }
18  ]
```

Bước 14

Tạo file Demo03.js để thực hiện Get all dữ liệu trong file data.json và public thành API cho Front-End có mã nguồn như sau:

ExpressJS

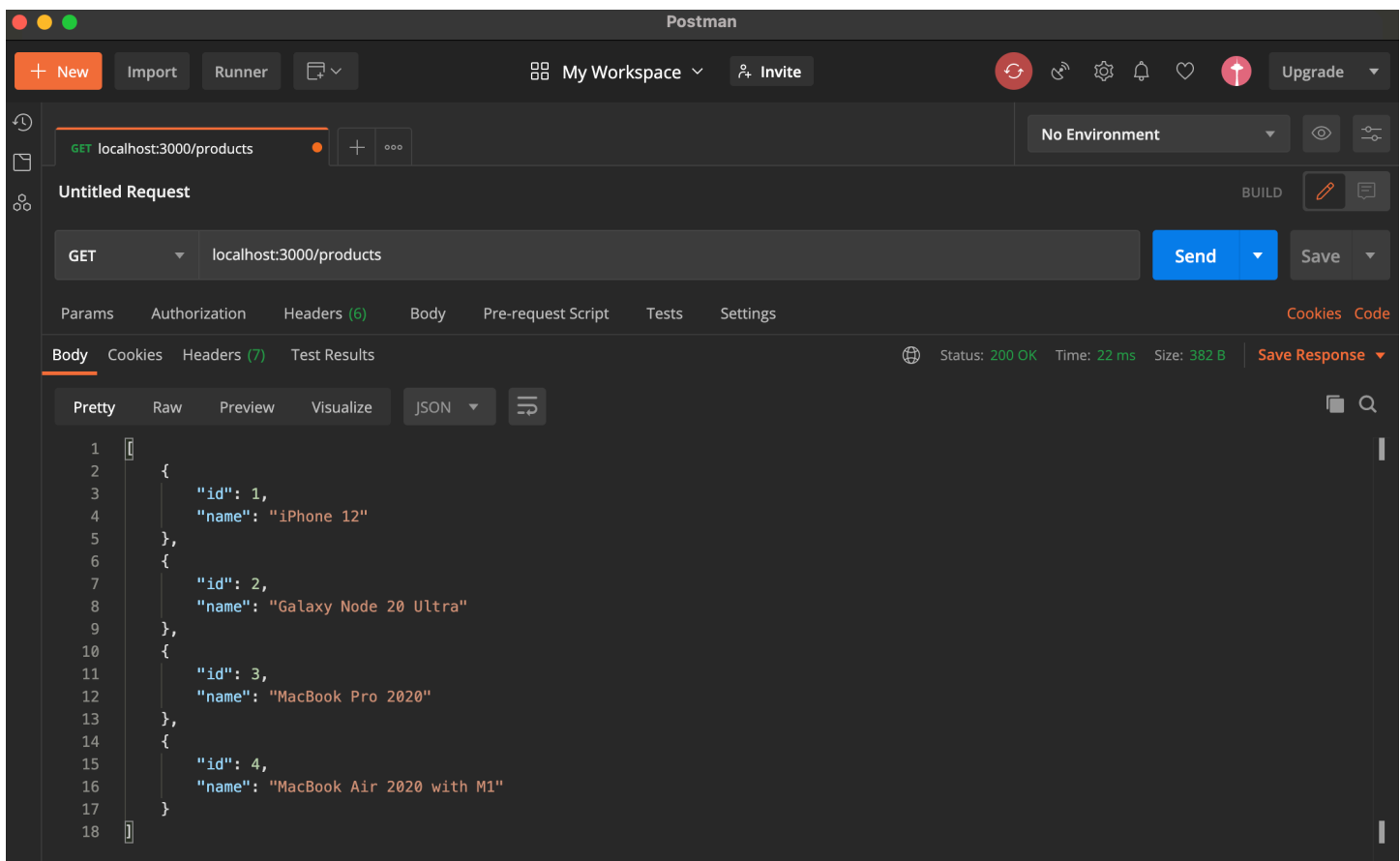
```
JS Demo03.js > ...
1  const express = require('express')
2
3  var products = require('./data.json')
4
5  const app = express()
6  const port = 3000
7
8  app.listen(port, () => {
9    console.log(`Example app listening at http://localhost:\${port}`)
10 })
11
12 app.get('/products', (req, res) => {
13   res.status = 200
14   res.setHeader('Content-Type', 'application/json')
15   res.send(products)
16 })
```

Sử dụng lệnh `$ node Demo03.js` để Deploy

Bước 15

Sử dụng Postman để truy cập vào url: <http://localhost:3000/products> để lấy dữ liệu của file **Data.json**

ExpressJS



Bước 16

Trong file Demo03.js, bổ sung thêm API **/products/:id** có mã nguồn như sau:

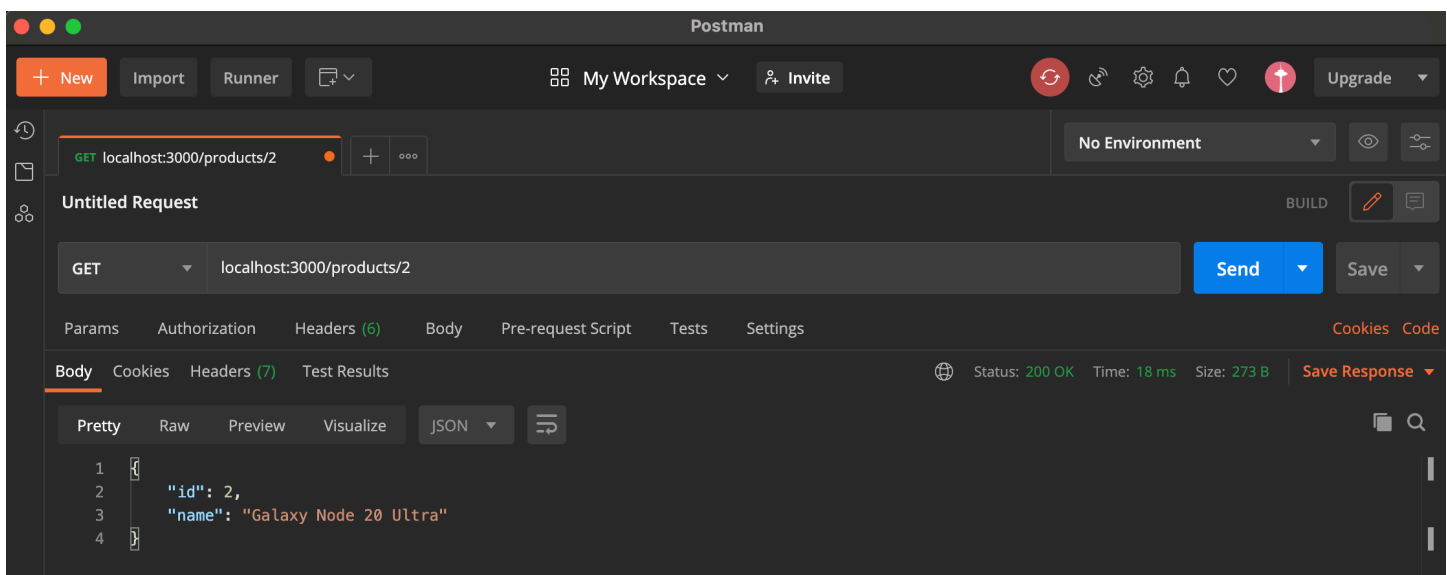
ExpressJS

```
18 app.get('/products/:productID', (req, res) => {
19     let id = req.params['productID']
20
21     res.status = 200
22     res.setHeader('Content-Type', 'application/json')
23     res.send(products[id - 1])
24 })
```

Sử dụng lệnh `$ node Demo03.js` để Deploy

Bước 15

Sử dụng Postman để truy cập vào url: <http://localhost:3000/products/2> để lấy dữ liệu của product có ID là 2 trong file data.json



Hết

Chúc các bạn làm bài tốt