**ỦY BAN NHÂN DÂN TP.HCM**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**BÁO CÁO ĐỒ ÁN CHUYÊN ĐỀ WEB 2**

**NĂM HỌC 2018 - 2019**

**Tên đề tài: Xây dựng RESTful API**

**Giảng viên: Phan Thanh Nhuần**

**Nhóm trưởng: Phạm Ngọc Huy Duy (16211TT1061)**

**Nhóm phó: Nguyễn Xuân Sơn (16211TT3147)**

**Thành viên 1: Nguyễn Anh Dũng (16211TT0420)**

**Thành viên 2: Phạm Vũ Quang (16211TT0866)**

**TP. Hồ Chí Minh, năm 2019**

**ỦY BAN NHÂN DÂN TP.HCM**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**BÁO CÁO ĐỒ ÁN CHUYÊN ĐỀ WEB 2**

**NĂM HỌC 2018 - 2019**

**Tên đề tài: Xây dựng RESTful API**

**Giảng viên: Phan Thanh Nhuần**

**Nhóm trưởng: Phạm Ngọc Huy Duy (16211TT1061)**

**Nhóm phó: Nguyễn Xuân Sơn (16211TT3147)**

**Thành viên 1: Nguyễn Anh Dũng (16211TT0420)**

**Thành viên 2: Phạm Vũ Quang (16211TT0866)**

**TP. Hồ Chí Minh, năm 2019**

MỤC LỤC

[LỜI GIỚI THIỆU 2](#_bookmark0)

[CHƯƠNG1: RESTful API 3](#_bookmark1)

[CHƯƠNG 2: PASSPORT](#_bookmark2) 5

1. [Passport là gì?](#_bookmark3) 5
2. [Hướng dẫn cài đặt passport trong laravel](#_bookmark4) 6

[CHƯƠNG 3: ĐẶC TẢ YÊU CẦU 1](#_bookmark5)4

1. [Mô tả tính năng 1](#_bookmark6)4
2. [Các API đã viết 1](#_bookmark7)4

[CHƯƠNG 4: BÁO CÁO CÔNG VIỆC](#_bookmark12) 15

1. [Phạm Ngọc Huy Duy 1](#_bookmark6)5
2. [Nguyễn Xuân Sơn 1](#_bookmark7)6
3. [Nguyễn Anh Dũng 1](#_bookmark7)7
4. [Phạm Vũ Quang 1](#_bookmark7)8

# LỜI GIỚI THIỆU

Như ở Chuyên đề Web 1 đã giới thiệu về Laravel, thì sang Chuyên đề Web 2 nhóm sẽ dựa trên Laravel để viết API bằng Passport

Hiện nay có rất nhiều ngôn ngữ lập trình hoặc framework hỗ trợ cho việc thiết kế API cho ứng dụng web nhưng theo việc sử dụng và hỗ trợ mạnh cho việc thiết kế API thì Passport gần như là sự lựa chọn của đa số các lập trình viên.

Chúng em xin chân thành cảm ơn thầy Phan Thanh Nhuần đã hướng dẫn chúng em hoàn thành cả hai chuyên đề phát triển web trong kì này.

# CHƯƠNG 1: RESTful API

Quản lý resource (tài nguyên) là một phần quan trọng và chiếm phần lớn trong việc phát triển website. Trong đó resource của các website khác nhau có thể sẽ khác nhau**.**



*Hình 1: RESTful API design*

Việc quản lý resource của một website bao gồm 4 tác vụ chính:

* Tạo mới một resource (create)
* Lấy thông tin một resource (read)
* Cập nhật một resource (update)
* Xoá một resource (delete)

## **RESTful API Là Gì**

RESTful API là một tiêu chuẩn dùng trong việc thết kế các thiết kế API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến nhất ngày nay.

RESTful API là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức kết nối với các thư viện ứng dụng khác.

Trọng tâm của REST quy định cách sử dụng các HTTP method (như *GET*, *POST*, *PUT*, *DELETE*...) và cách định dạng các URL cho ứng dụng web để quản các resource. Ví dụ với một trang blog để quản lý các bài viết chúng ta có các URL đi với HTTP method như sau:

* URL tạo bài viết: http://my-blog.xyz/posts. Tương ứng với HTTP method là POST
* URL đọc bài viết với ID là 123: http://my-blog.xyz/posts/123. Tương ứng với HTTP method là GET
* URL cập nhật bài viết với ID là 123: http://my-blog.xyz/posts/123. Tương ứng với HTTP method là PUT
* URL xoá bài viết với ID là 123: http://my-blog.xyz/posts/123. Tương ứng với HTTP method là DELETE

Tất cả các request đã thực hiện đều sẽ có mã trạng thái HTTP (HTTP status code) và được chia thành 5 lớp:

* 1xx: Hàm có nghĩa mang thông tin
* 2xx: Hàm có nghĩa thành công
* 3xx: Hàm có nghĩa điều hướng
* 4xx: Hàm có nghĩa lỗi từ phía client side
* 5xx: Hàm có nghĩa có lỗi từ máy chủ

Tổng hợp các khái niệm trên lại chúng ta có RESTful API là các API tuân thủ theo REST.

## **REST và REST(ful)**

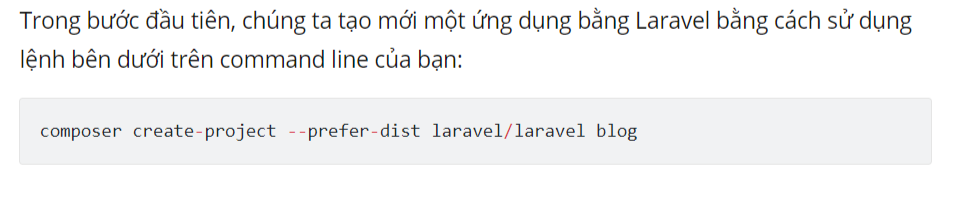
Một vấn đề mà nhiều bạn mới tìm hiểu về RESTful cũng thường cảm thấy bối rối đó là RESTvà RESTful khác nhau như thế nào. REST là viết tắt của cụm từ Representational State Transfer và các ứng dụng sử dụng kiểu thiết kế REST thì được gọi là RESTful (-ful là tiếp vị ngữ giống như beauty và beautiful). Tất nhiên bạn cũng có thể sử dụng thuật ngữ REST thay cho RESTful và ngược lại.

# CHƯƠNG 2: PASSPORT

Với sự gia tăng của phát triển phần mềm mobile và JavaScript, sử dụng API RESTful là lựa chọn tốt nhất để xây dựng một giao diện duy nhất giữa dữ liệu của bạn và khách hàng của bạn (gọi tắt là API).

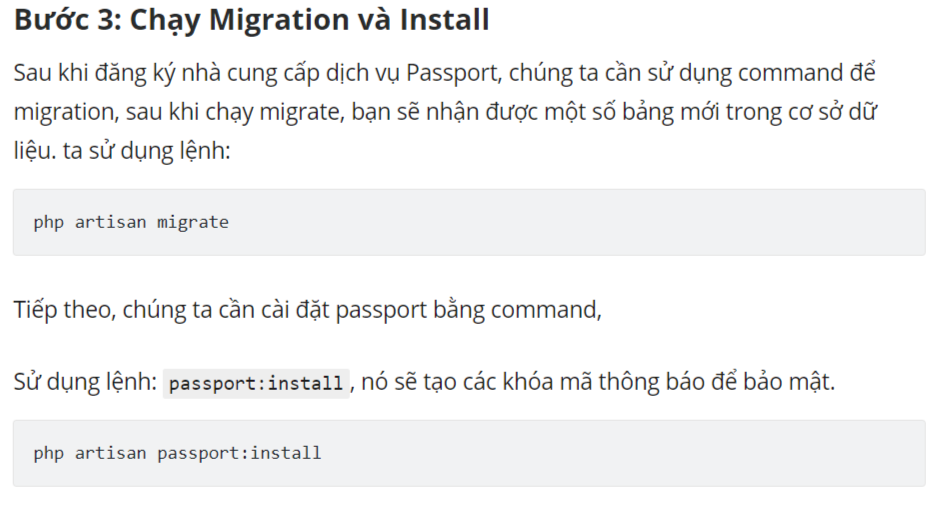
Laravel là một framework PHP rất đơn giản, đa dạng và có thể hỗ trợ người lập trình tiết kiệm được rất nhiều thời gian với các sự tiện dụng thông qua các cấu hình. Framework này cũng nhằm mục đích phát triển với web và đã kết hợp nhiều tính năng và ý tưởng mới trong thế giới phát triển web - chẳng hạn như hàng đợi công việc, xác thực API, giao tiếp trong thời gian thực và nhiều hơn nữa.

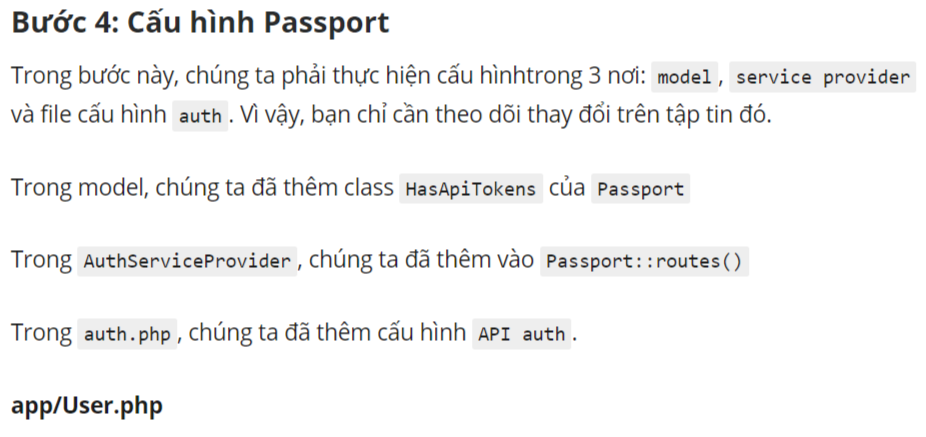
**Bước 1: Cài đặt Laravel**



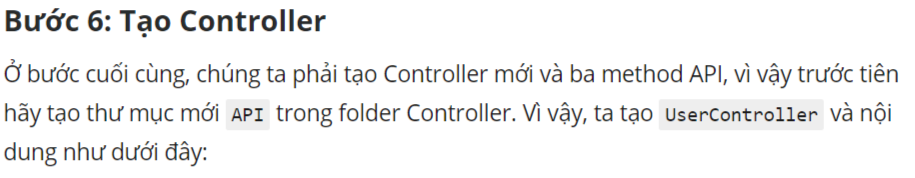
### Bước 2: Cài đặt Package

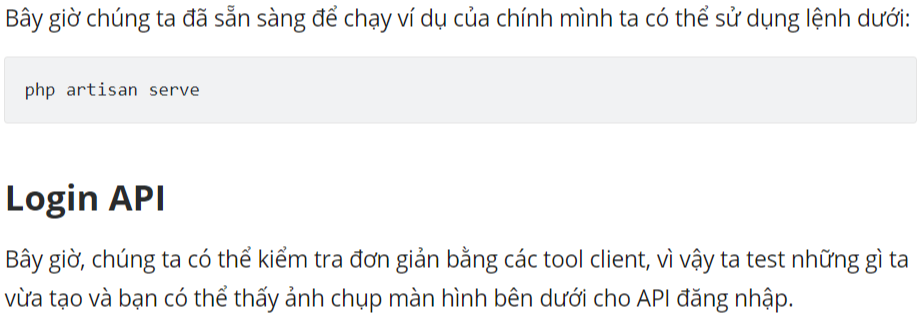


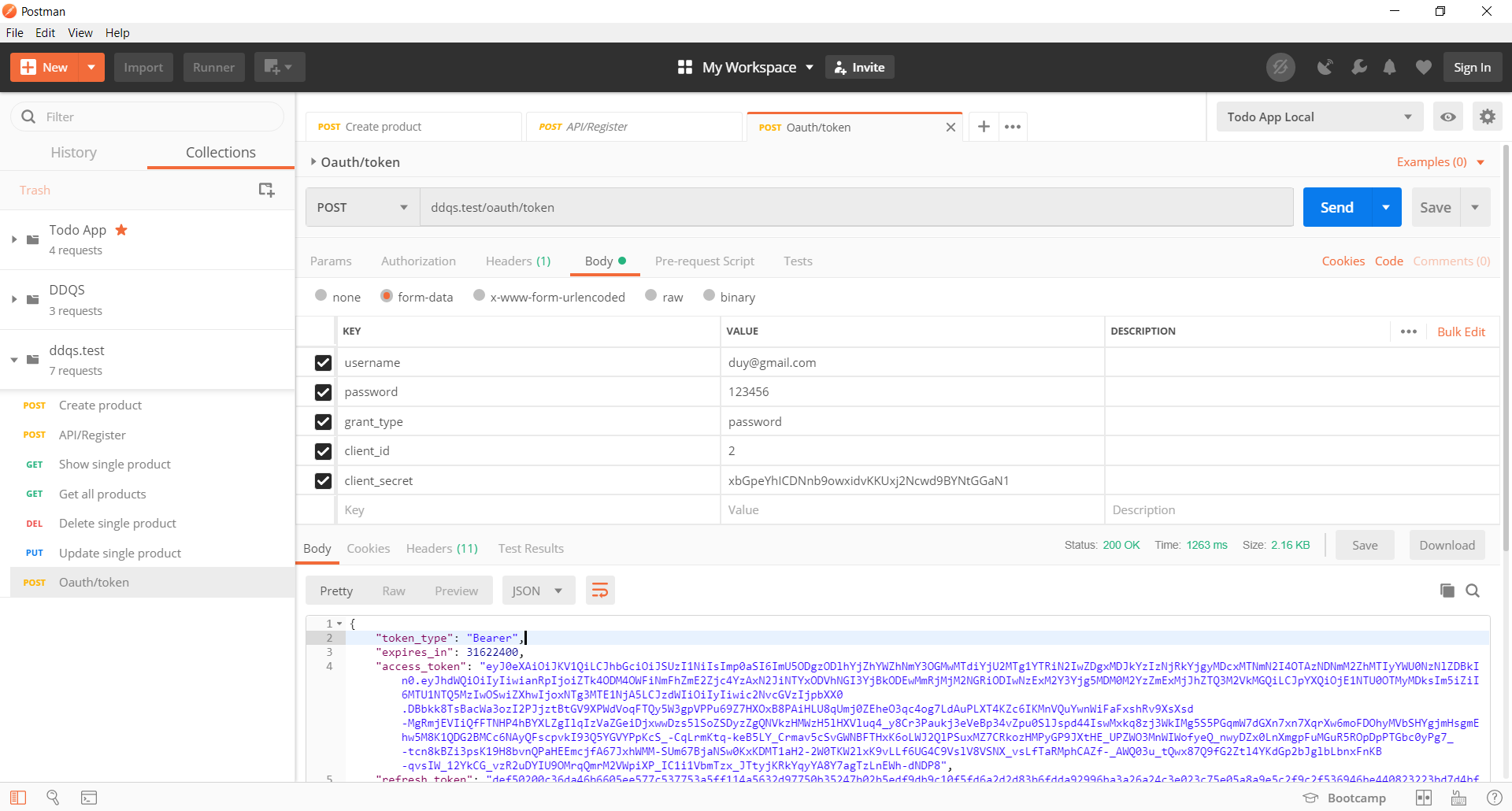




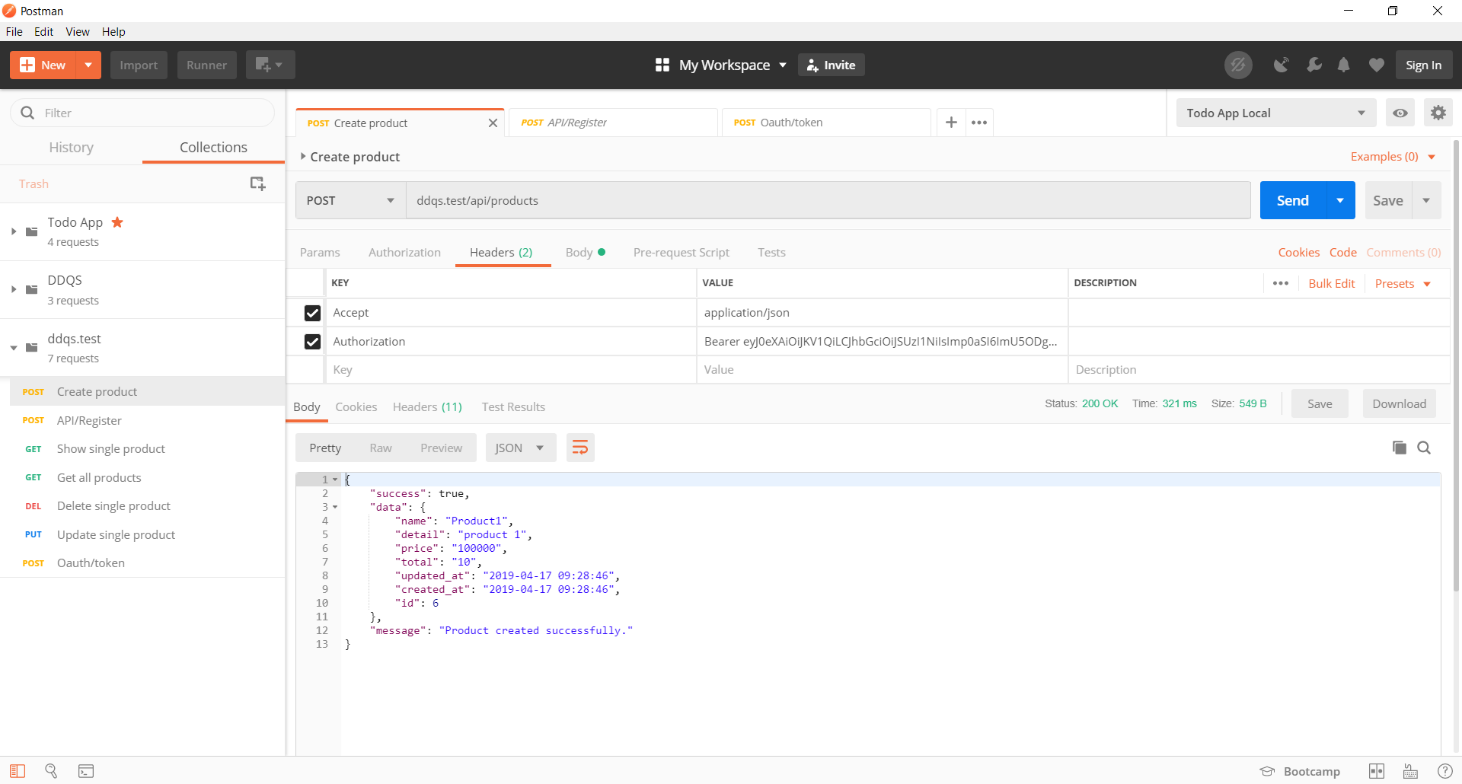




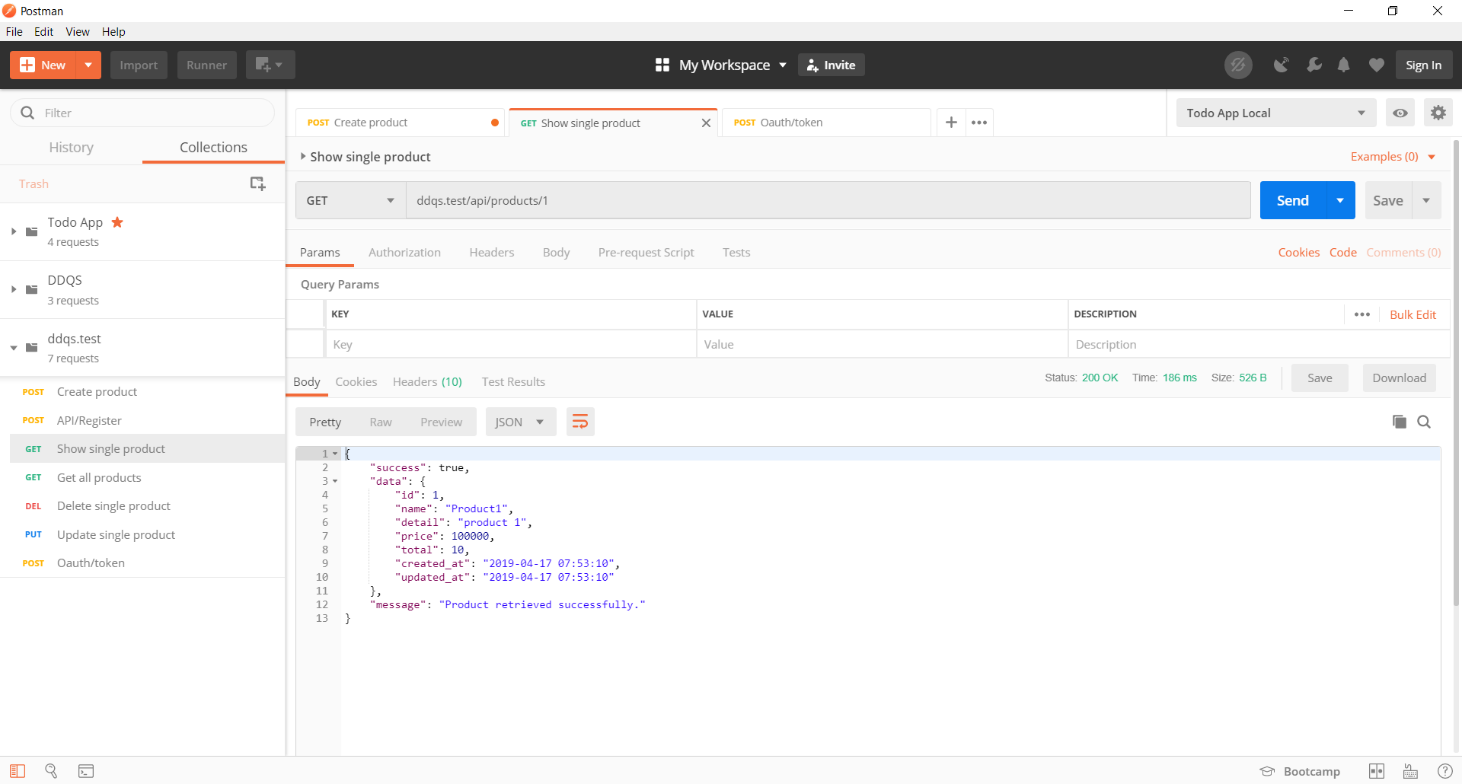




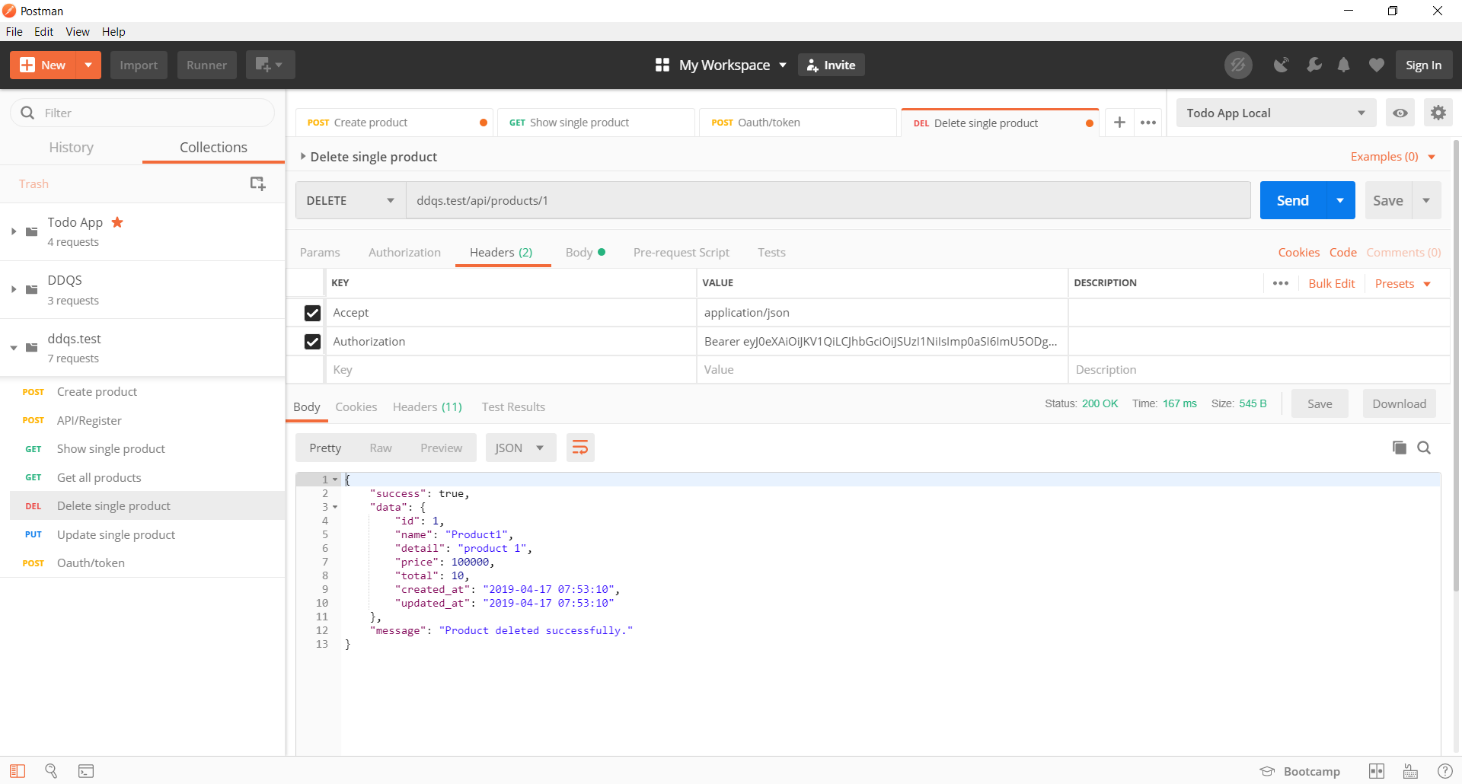
*Hình 1: get token*



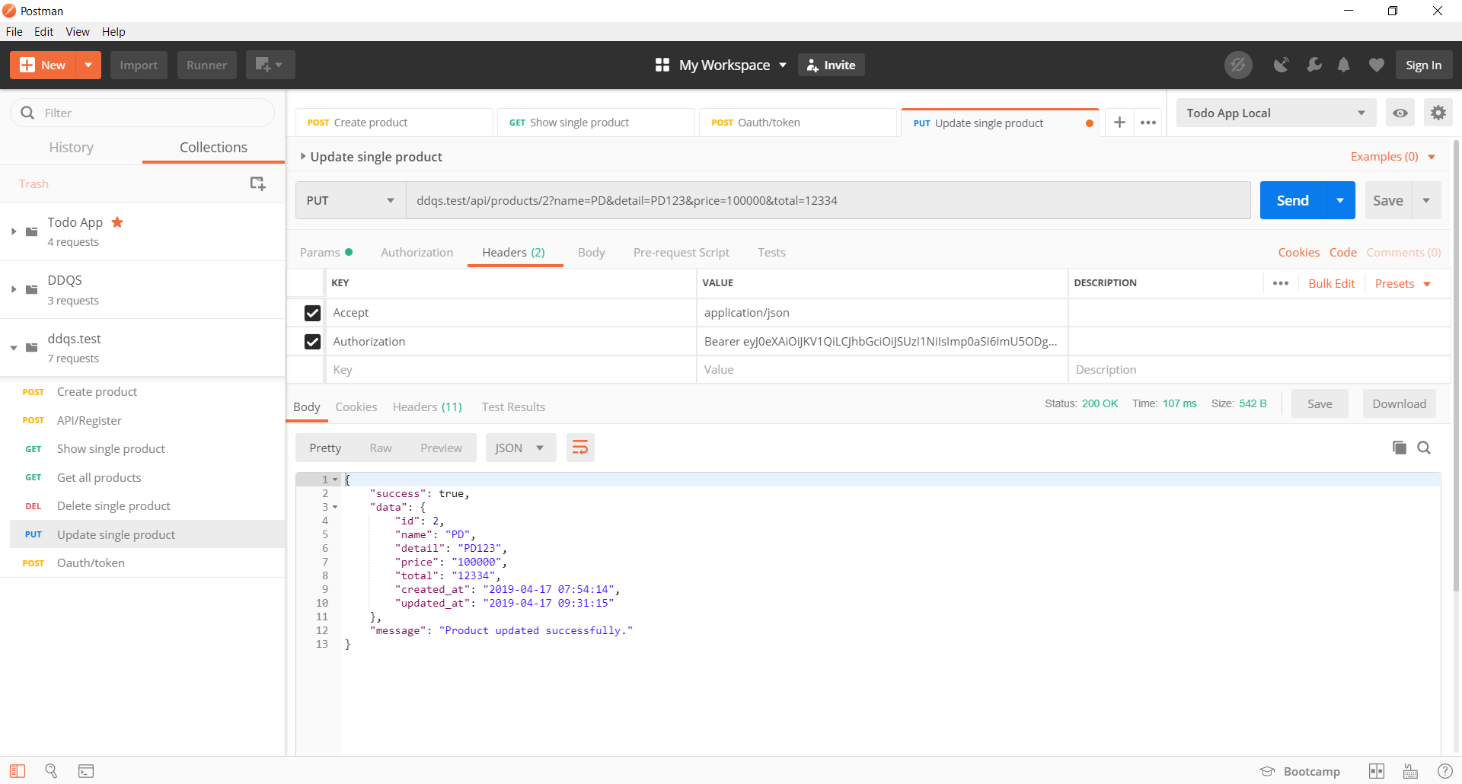
*Hình 2: Create product*



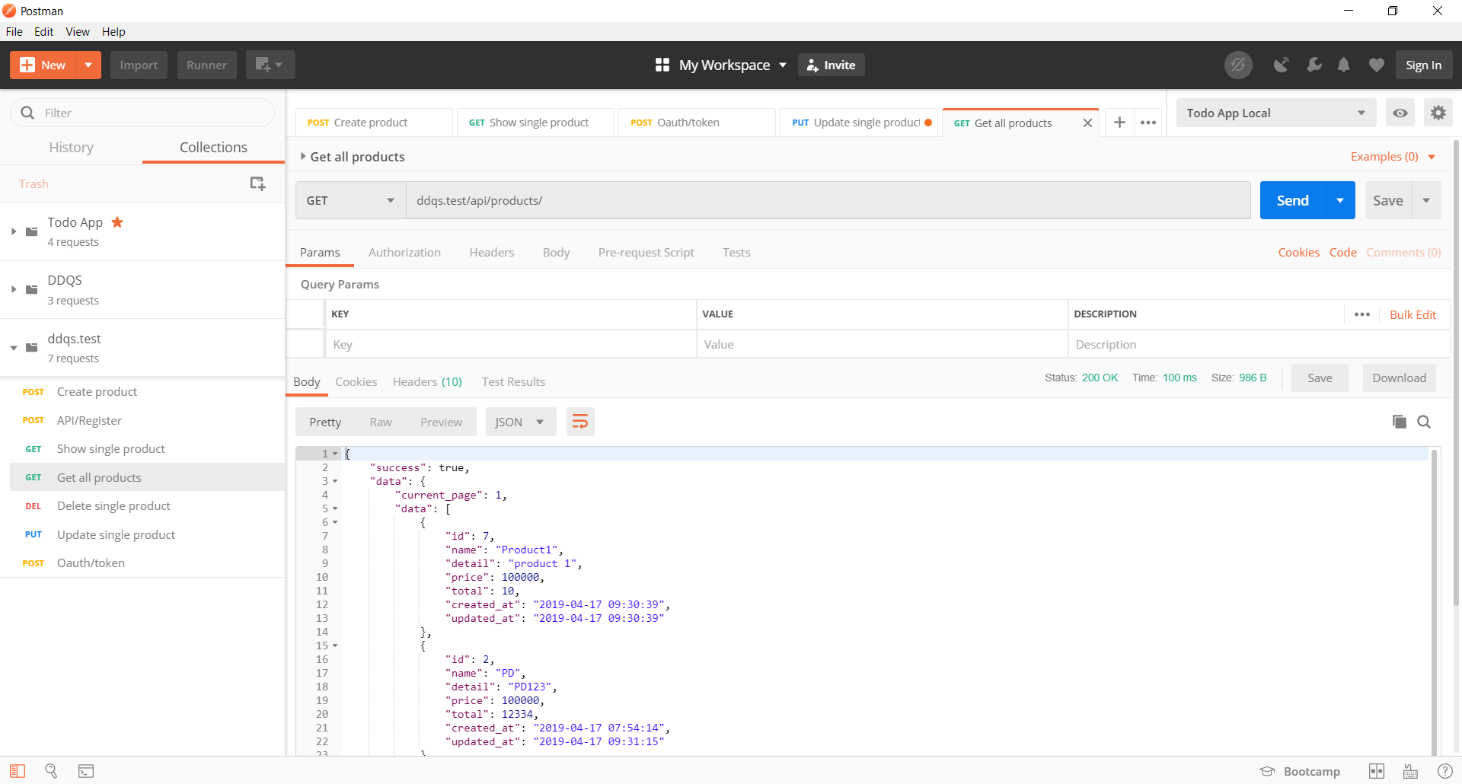
*Hình 3: show single product*



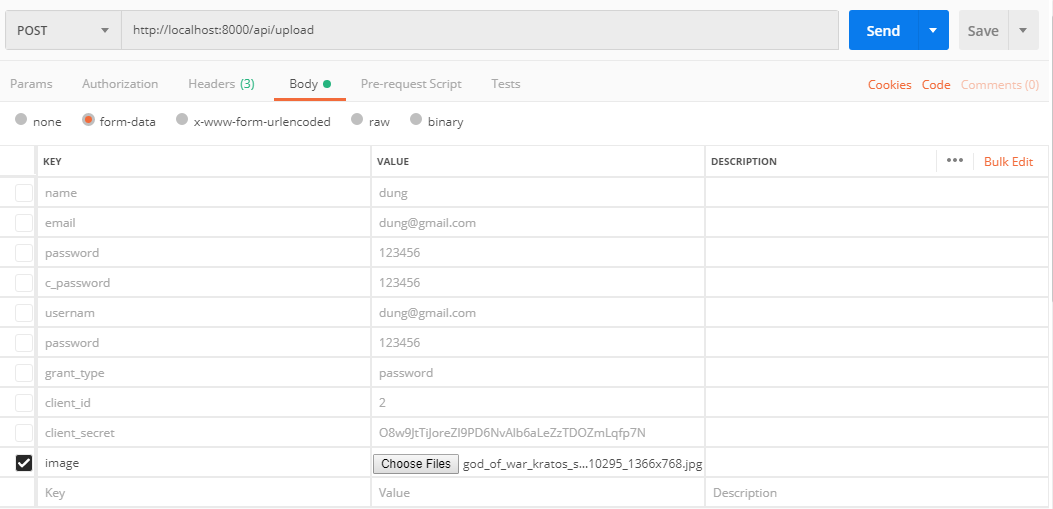
*Hình 4: delete single product*



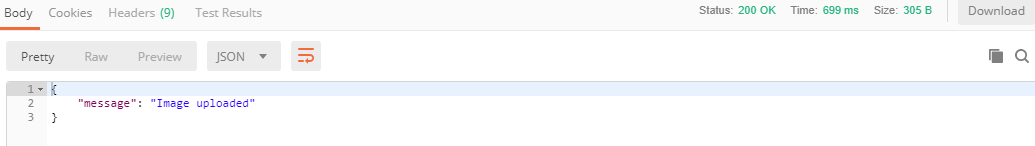
*Hình 5: update single product*



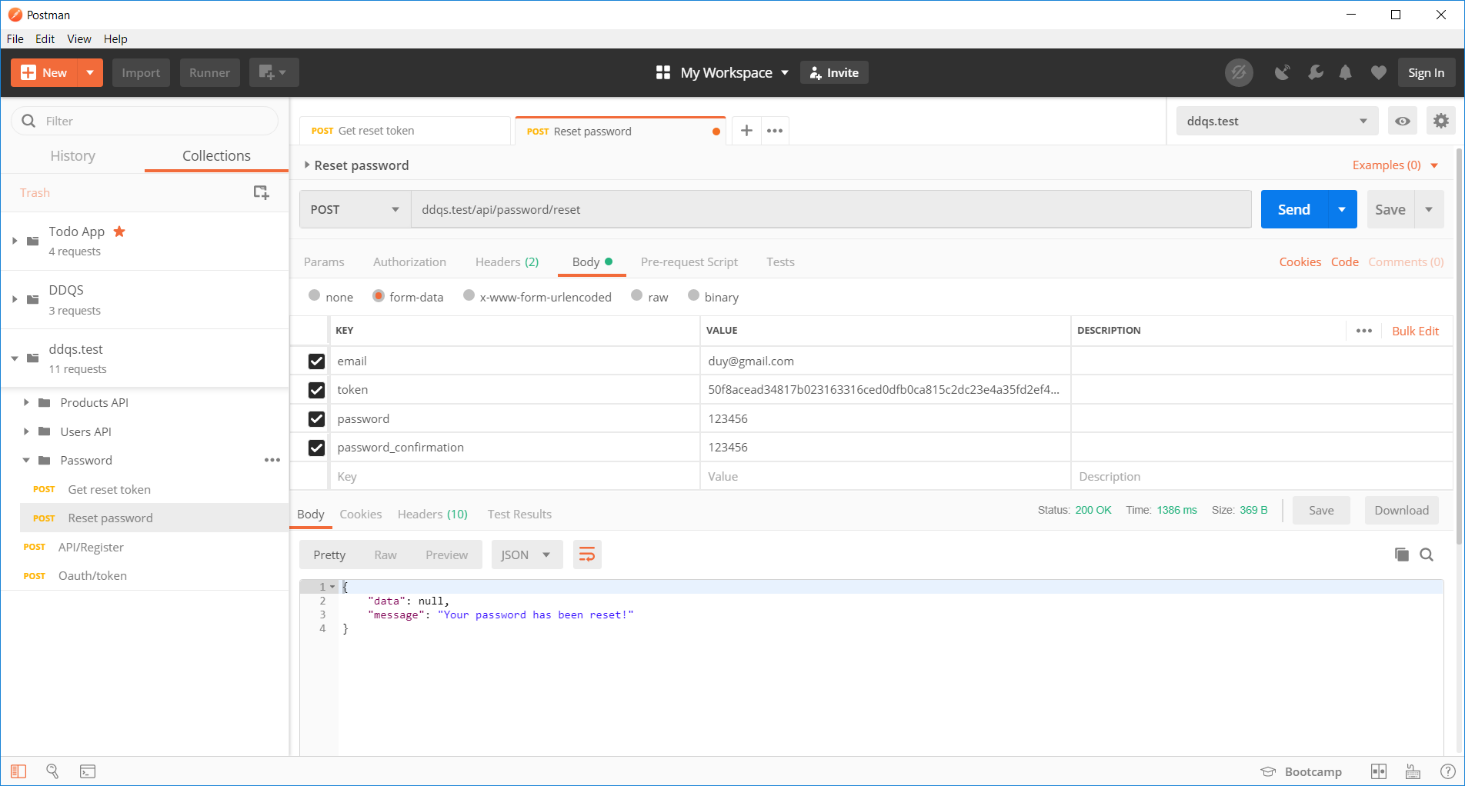
*Hình 6: show all procduct*



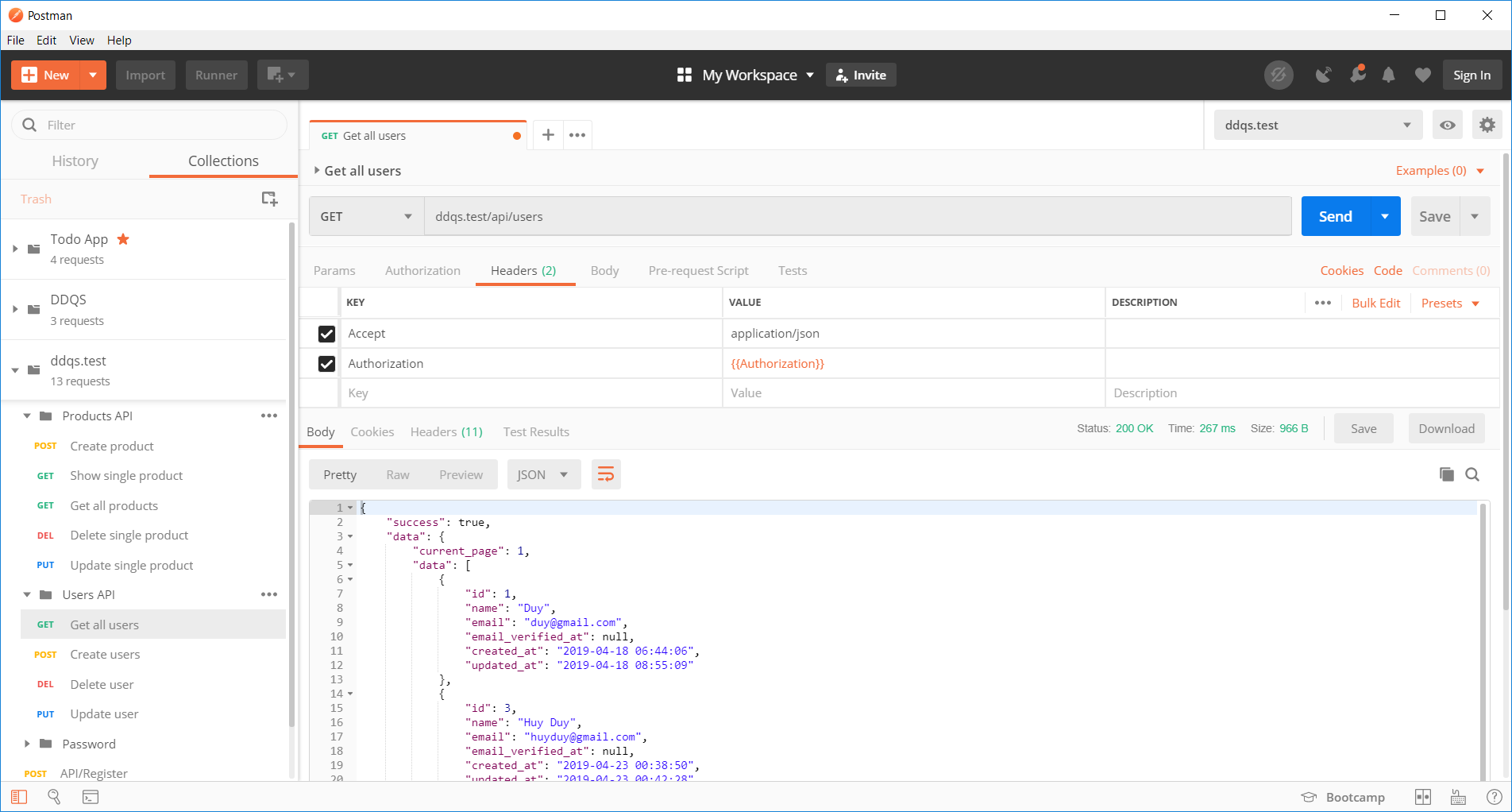
*Hình 7: upload*



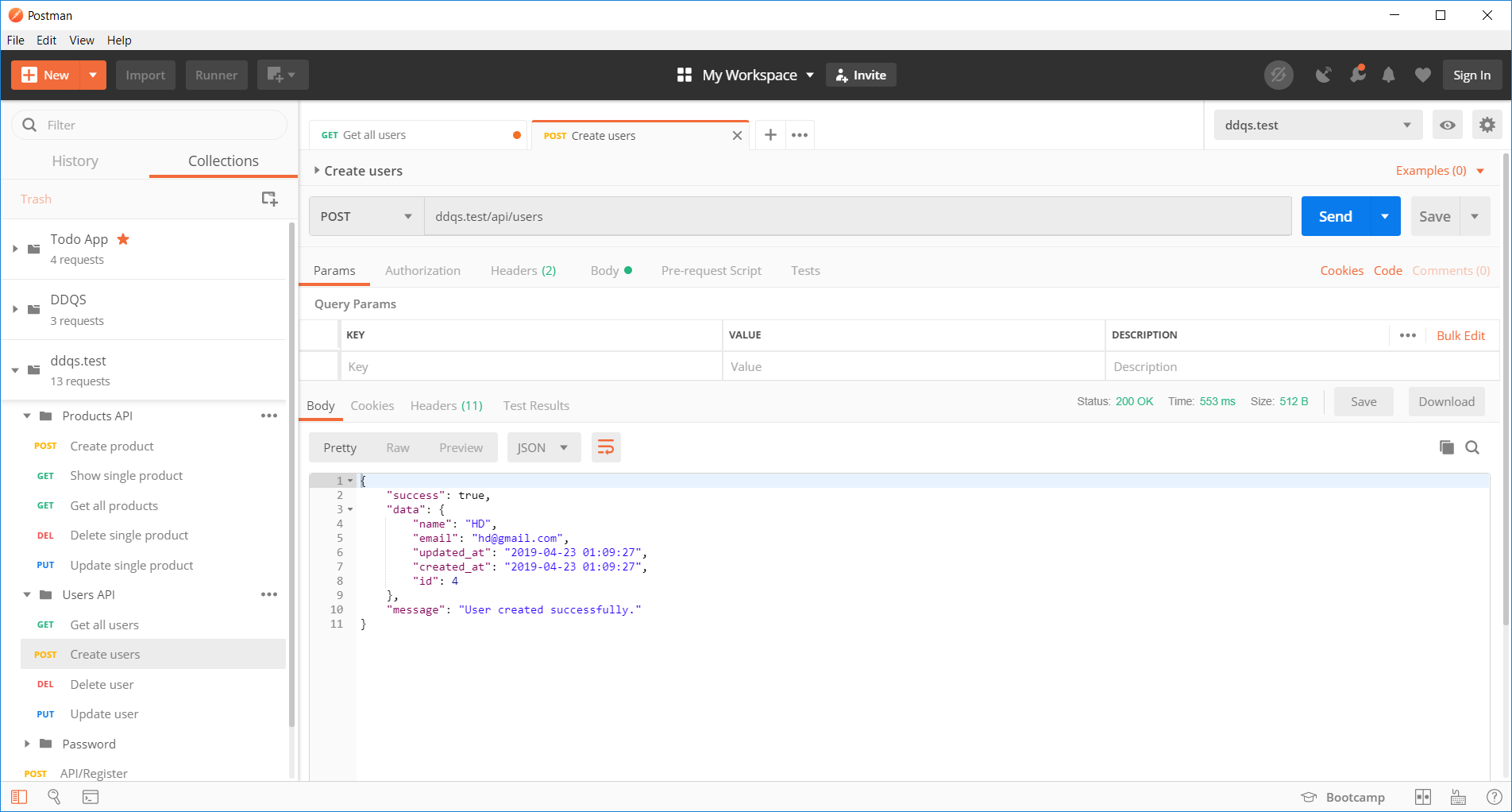
*Hình 8: show message image complete!*

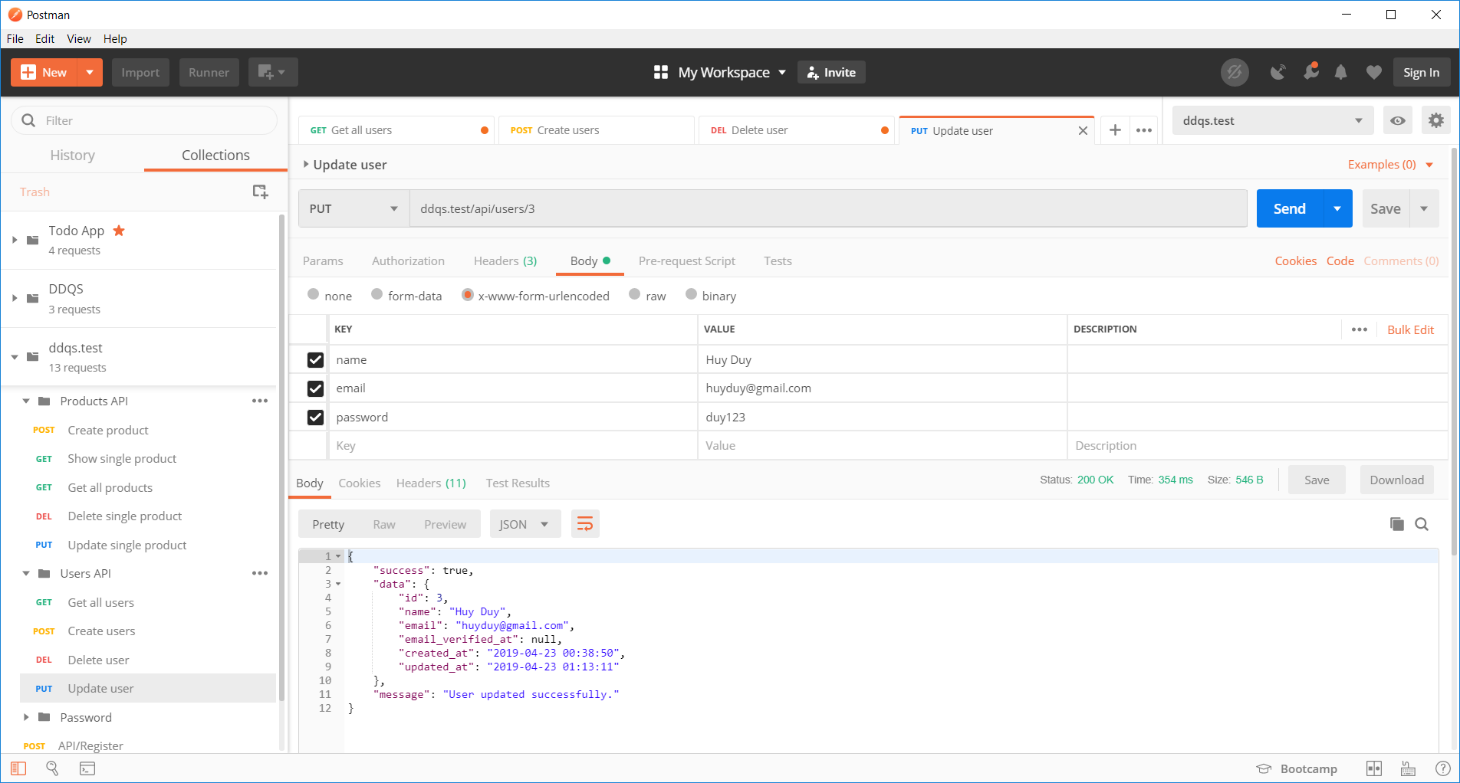
**

*Hình 9: show reset password*

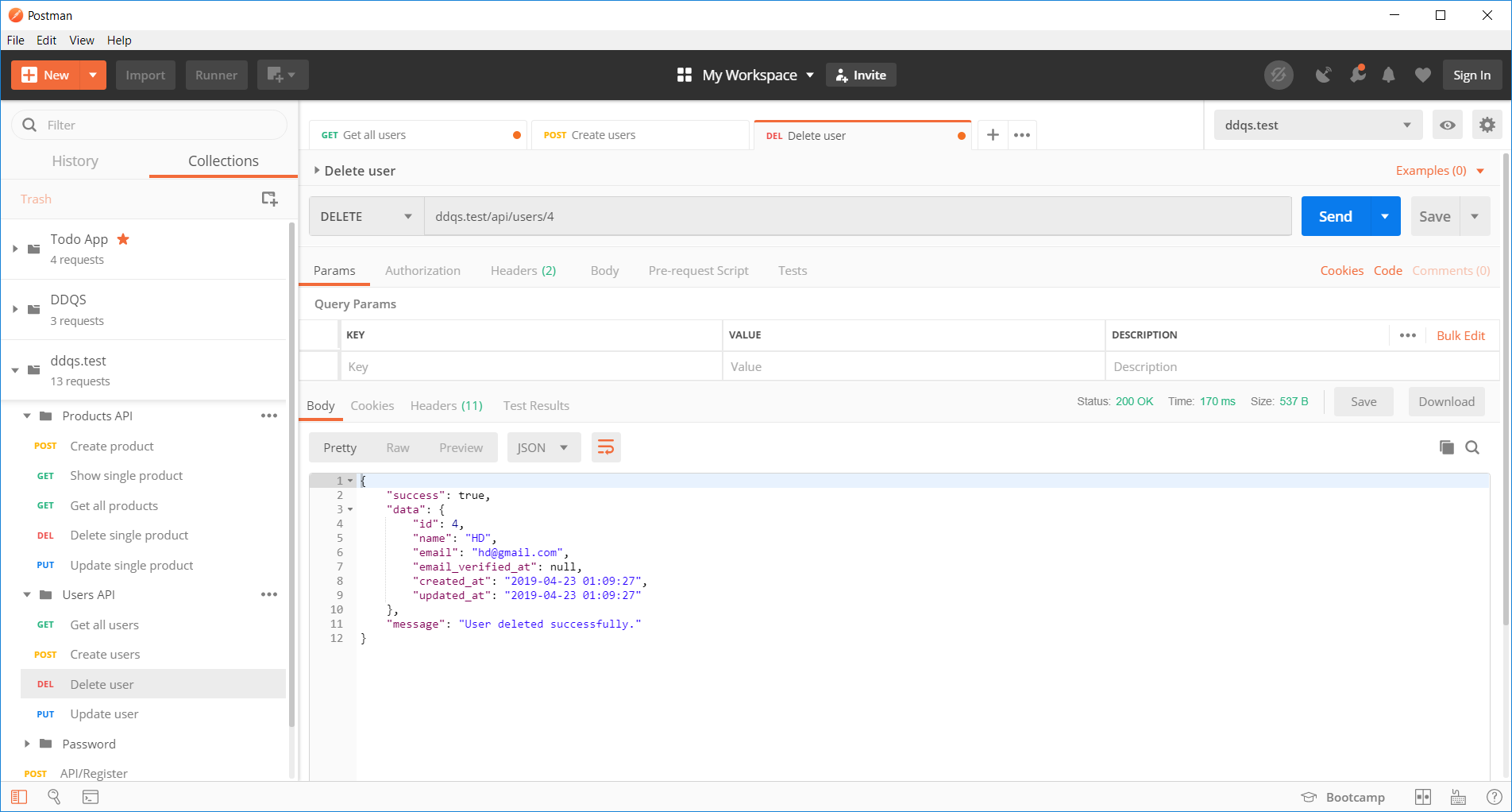


*Hình 10: Show all users*



*Hình 11: Create a user*

*Hình 12: Update a user*



*Hình 13: Delete a user*

# CHƯƠNG 3: ĐẶC TẢ YÊU CẦU

## Mô tả tính năng

## Sản phẩm

* + - * Thêm thông tin sản phẩm
      * Sửa thông tin sản phẩm
      * Xóa thông tin sản phẩm
      * Hiển thị thông tin sản phẩm

Người dùng

* + - * Thêm thông tin người dùng
      * Sửa thông tin người dùng
      * Xóa thông tin người dùng
      * Hiển thị thông tin người dung

Upload hình ảnh

Reset password

## Các API đã viết

|  |  |  |
| --- | --- | --- |
| Method | URI | Name |
| POST | Api/password/email | Get reset token |
| POST | Api/password/reset | Reset password if token is valid |
| POST | Api/products | Create product |
| GET | Api/products | Get all products and pagination |
| DELETE | Api/products/{id} | Delete a product |
| PUT|PATCH | Api/products/{id} | Update a single product |
| POST | Api/users | Create user |
| GET | Api/users | Get all users and pagination |
| DELETE | Api/users/{id} | Delete a user |
| PUT|PATCH | Api/users/{id} | Update a user product |
| POST | Oauth/token | Get access\_token to use protected API |

# CHƯƠNG 4: BÁO CÁO CÔNG VIỆC

1. Phạm Ngọc Huy Duy

a. Get reset password token

- URL: api/password/email

- Description: Get reset password token by given email

- Request method: POST

- Request parameter:

+ email

- Request result:

Success:

* Header: responsive status: 200 OK
* Body: Show token for reset password

Fail

* Header: responsive status: 400 Bad Request
* Body: Show message “Email not found.”

b. Reset password

URL: api/password/reset

- Description: Reset password if the token is valid

- Request method: POST

- Request parameter:

+ email

+token

+password

+password\_confirmation

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “Your password has been reset!”

Fail

* Header: responsive status: 404
* Body: Show message “This password reset token is invalid.”

c. Create Oauth token

URL: oauth/token

- Description: Create access\_token to use protected API

- Request method: POST

- Request parameter:

+username

+password

+grant\_type

+client\_id

+client\_secret

- Request result:

Success:

* Header: responsive status: 200
* Body: Show results with access\_token

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “The user credentials were incorrect.”

2. Nguyễn Xuân Sơn

a. Show single product

- URL: api/products/id?

- Description: show single product

- Request method: GET

- Header: header authorization

- Request parameter:

+ ID

+ Name

+ Detail

+ Price

+ Total

- Request result:

Success:

* Header: responsive status: 200
* Body: Show single product and message “Product update successfully”

Fail

* Header: responsive status: 404 not found
* Body: Show message “Product not found”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

b. Delete single product

URL: api/products/id?

- Description: delete single product

- Request method: DELETE

- Header: header authorization

- Request parameter:

+ ID

+ Name

+ Detail

+ Price

+ Total

+ Created at

+ Update at

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “Product delete successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

c. Update users

URL: api/users/1

- Description: Update users

- Request method: PUT

- Header: header authorization

- Request parameter:

+ ID

+ Name

+ Email

+ Password

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “User updated successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

3. Nguyễn Anh Dũng

a. Create product

- URL: api/products

- Description: Create product

- Request method: POST

- Header: header authorization

- Request parameter:

+ ID

+ Name

+ Detail

+ Price

+ Total

- Request result:

Success:

* Header: responsive status: 200
* Body: Show single product and message “Product update successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

b. Show all product

- URL: api/products

- Description: Show all product

- Request method: GET

- Header: header authorization

- Request parameter:

+ ID

+ Name

+ Detail

+ Price

+ Total

- Request result:

Success:

* Header: responsive status: 200
* Body: Show single product and message “Product update successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

c. Upload image

URL: api/upload

- Description: delete single product

- Request method: POST

- Header: header authorization

- Request parameter:

+ ID

+ Image

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “Image uploaded”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

4. Phạm Vũ Quang

a. Update single product

- URL: api/products/2

- Description: Update single product

- Request method: PUT

- Request parameter:

+ ID

+ Name

+ Detail

+ Price

+ Total

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “Product updated successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated”

b. Get all users and pagination

URL: api/users

- Description: Get all users

- Request method: GET

- Header: header authorization (Bearer)

- Request parameter:

- Request result:

Success:

* Header: responsive status: 200
* Body: Show data results and pagination

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated.”

c. Delete user

URL: api/users/1

- Description: Delete users

- Request method: DELETE

- Header: header authorization (Bearer)

- Request parameter:

- Request result:

Success:

* Header: responsive status: 200
* Body: Show message “User delete successfully”

Fail

* Header: responsive status: 401 Unauthorized
* Body: Show message “Unauthenticated.”