

## LAPORAN PROGRES SERVER

Server dibangun dengan menggunakan beberapa teknologi sebagai berikut:

- Node.js -> sebagai runtime javascript
- Express.js -> sebagai framework backend
- MongoDB -> sebagai database
- JWT -> sebagai standar enkripsi autentikasi admin
- Cloudinary -> sebagai penyimpanan gambar produk
- Bcrypt.js -> sebagai enkripsi password admin
- Stripe -> sebagai dummy pembayaran online

Untuk progress pengerjaan server sudah selesai dari apa yang diinginkan oleh client dan juga admin. Dokumentasi lebih jelas mengenai request dan response silahkan kunjungi link dibawah ini untuk melihatnya.

<https://documenter.getpostman.com/view/18696549/2s8Z6yVsMZ>

Catatan: Install aplikasi postman bila diperlukan untuk melakukan import dokumentasi dan melakukan testing API.

### 1. Database Schema

#### A. Admin Schema

Digunakan untuk menyimpan data admin. Nantinya admin perlu login dahulu sebelum mereka dapat mengakses route yang di protect dengan beberapa role admin sebagai berikut:

Admin super Mereka memiliki izin berikut:

- Dapat melihat, membuat, memperbarui, dan menghapus pengguna admin.
- Dapat melihat, membuat, memperbarui, dan menghapus produk.
- Dapat melihat pesanan dan memperbarui statusnya.
- Dapat menghapus pesanan.

Admin moderate memiliki izin berikut:

- Dapat melihat, membuat, memperbarui, dan menghapus produk.
- Dapat melihat pesanan dan memperbarui statusnya.
- Dapat menghapus pesanan.

Admin low memiliki izin berikut:

- Dapat melihat pesanan dan memperbarui statusnya.

Field	Type	Required	Unique	Default
name	String	TRUE	FALSE	-
email	String	TRUE	TRUE	-
password	String	TRUE	FALSE	-
privilege	String	FALSE	FALSE	low

## B. Product Schema

Digunakan untuk menyimpan data dari produk yang nantinya akan disimpan kedalam database dengan keterangan schema seperti tabel dibawah ini.

Field	Type	Required	Unique	Default
name	String	TRUE	FALSE	-
description	String	TRUE	FALSE	-
price	Number	TRUE	FALSE	-
rating	Number	FALSE	FALSE	0
images	Object	TRUE	FALSE	-
colors	Array	TRUE	FALSE	-
company	String	TRUE	FALSE	-
stock	Number	TRUE	FALSE	-
numberOfReviews	Number	FALSE	FALSE	0
reviews	Object	FALSE	FALSE	-
shipping	Boolean	FALSE	FALSE	TRUE
featured	Boolean	FALSE	FALSE	FALSE
admin	Schema.Admin	TRUE	FALSE	-
createdAt	Date	FALSE	FALSE	Date.now()

## C. Order Schema

Digunakan untuk menyimpan data orderan dari user. Untuk schema dan keterangannya adalah sebagai berikut.

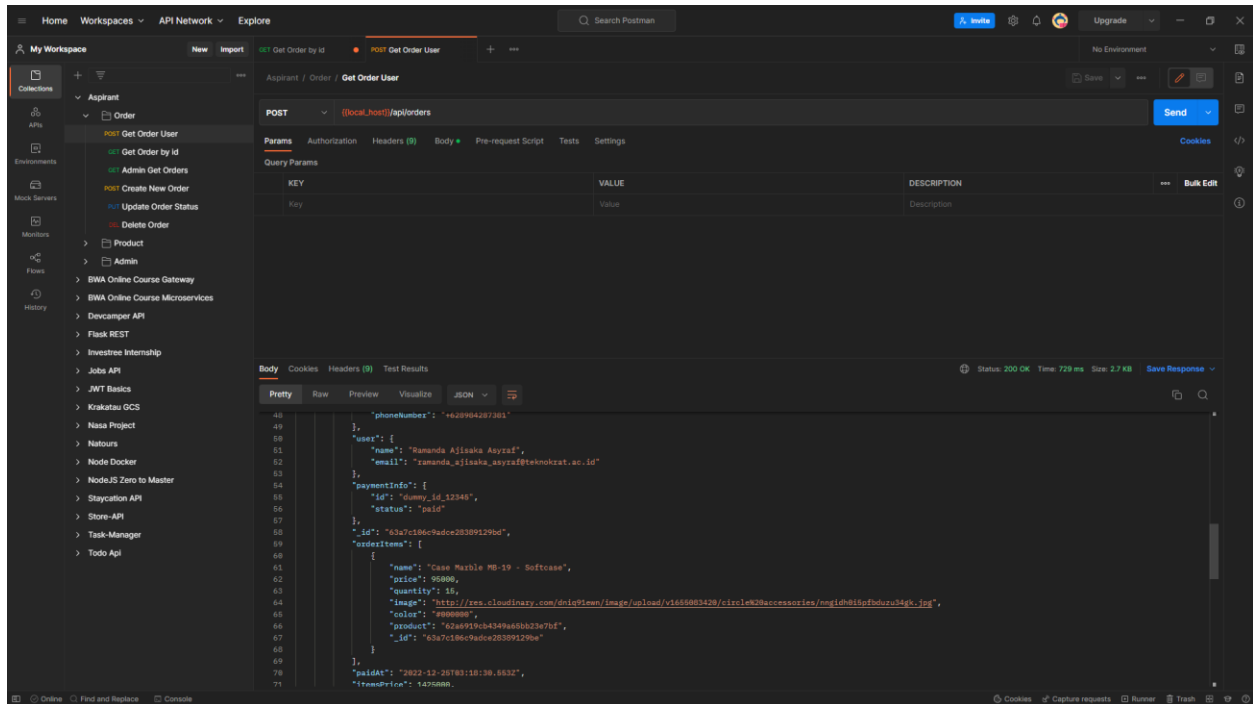
Field	Type	Required	Unique	Default
shippingInfo	Object	TRUE	FALSE	-
orderItems	Object	TRUE	FALSE	-
user	Object	TRUE	FALSE	-
paymentInfo	Object	TRUE	FALSE	-
paidAt	Date	TRUE	FALSE	-
itemsPrice	Number	TRUE	FALSE	0
shippingPrice	Number	TRUE	FALSE	0
totalPrice	Number	TRUE	FALSE	0
orderStatus	String	TRUE	FALSE	processing
createdAt	Date	FALSE	FALSE	Date.now()

deliveredAt	Date	FALSE	FALSE	-
-------------	------	-------	-------	---

## 2. Testing API

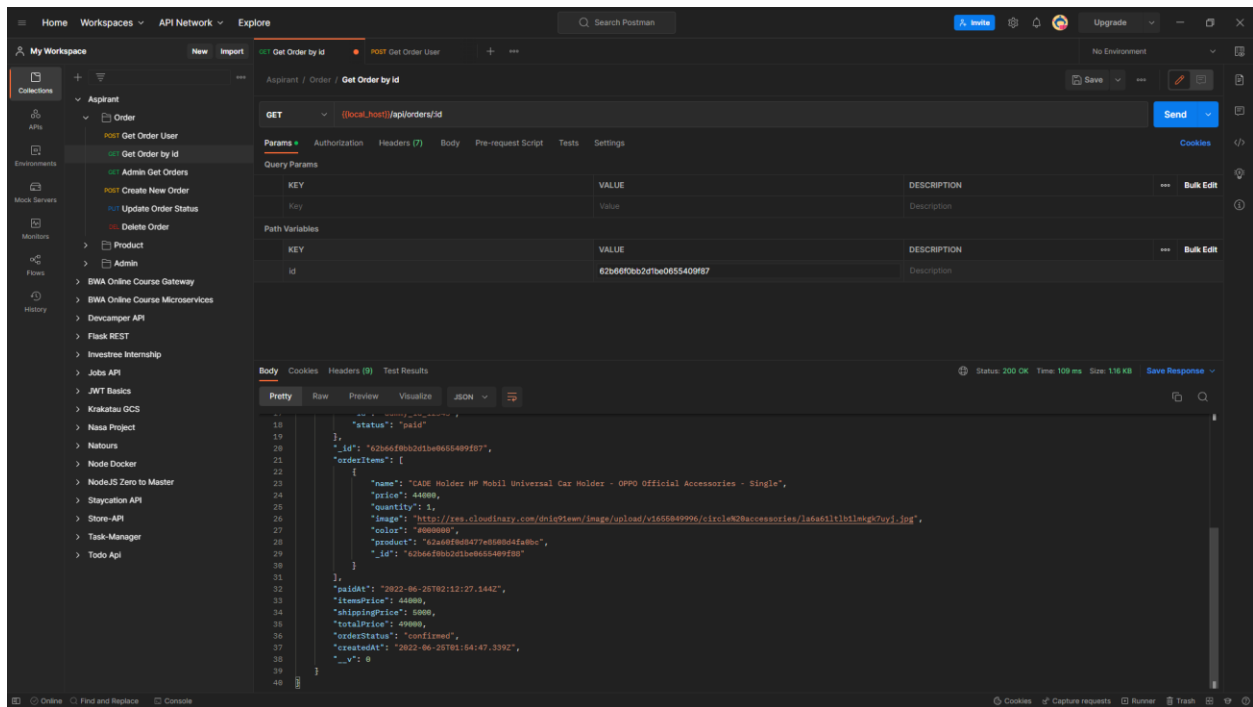
### A. Order Route

Get order users (host\_server/api/orders)



Endpoint api ini digunakan untuk mendapatkan seluruh detail order user berdasarkan email mereka. Pada saat melakukan request ke endpoint ini client harus mengirimkan body berupa email: “string email” dengan method POST.

## Get Order by Id (host\_server/api/orders/:id)

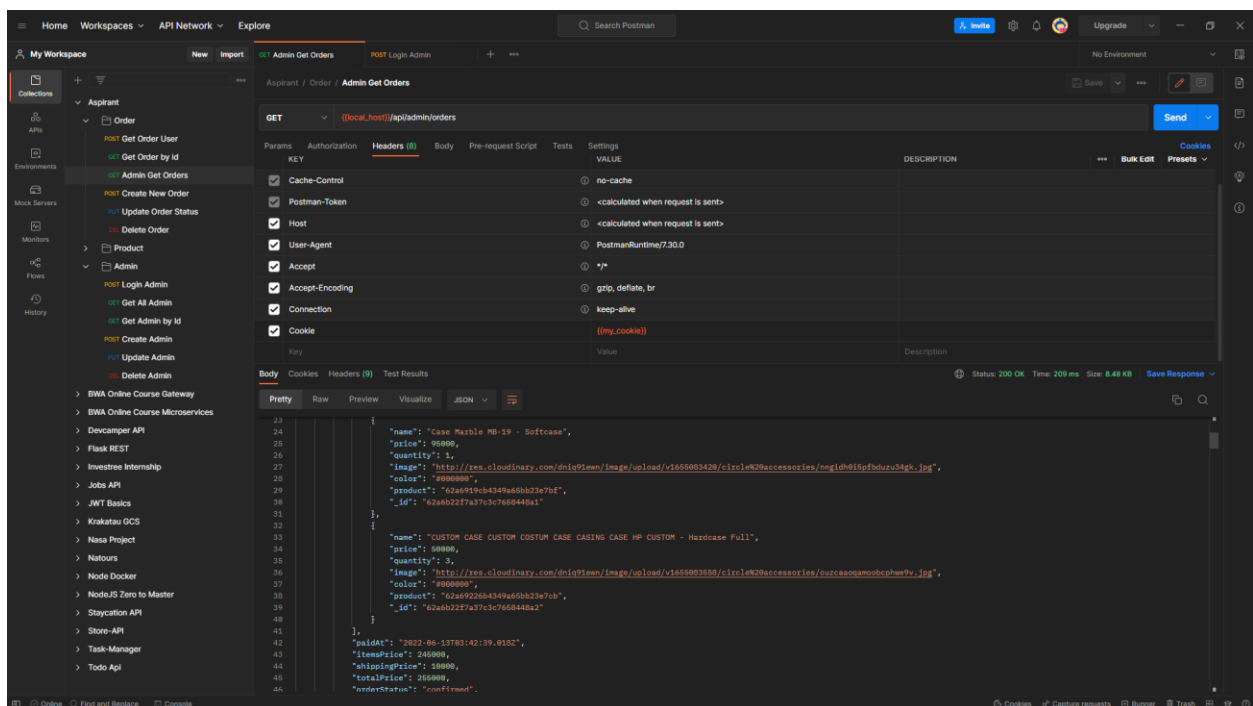


The screenshot shows a Postman workspace with a collection named 'Aspirant'. Inside the 'Order' folder, there is a GET request 'Get Order by Id' with the URL '[[local\_host]]/api/orders/:id'. The request is sent, and the response is displayed in the 'Body' tab. The response is a JSON object with the following structure:

```
{
  "status": "paid",
  "id": "62b6f0bb2d1be065409f07",
  "orderItems": [
    {
      "name": "CADE Holder HP Mobil Universal Car Holder - OPPO Official Accessories - Single",
      "price": 44800,
      "quantity": 1,
      "image": "http://res.cloudinary.com/dnig9iemw/image/upload/v1655849996/circleK20accessories/1skad11l1blmhp7uryj.jpg",
      "color": "#000000",
      "product": "62a6f0bb2d1be065409f07",
      "id": "62b6f0bb2d1be065409f07"
    }
  ],
  "paidAt": "2022-06-25T02:12:27.144Z",
  "itemPrice": 44800,
  "shippingPrice": 5000,
  "totalPrice": 49800,
  "orderStatus": "confirmed",
  "createdAt": "2022-06-25T01:54:47.339Z",
  "_v": 0
}
```

Endpoint api ini digunakan untuk mendapatkan order berdasarkan id dari order tersebut. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable berupa id orderan dengan method GET.

## Admin Get Order (host\_server/api/admin/orders)

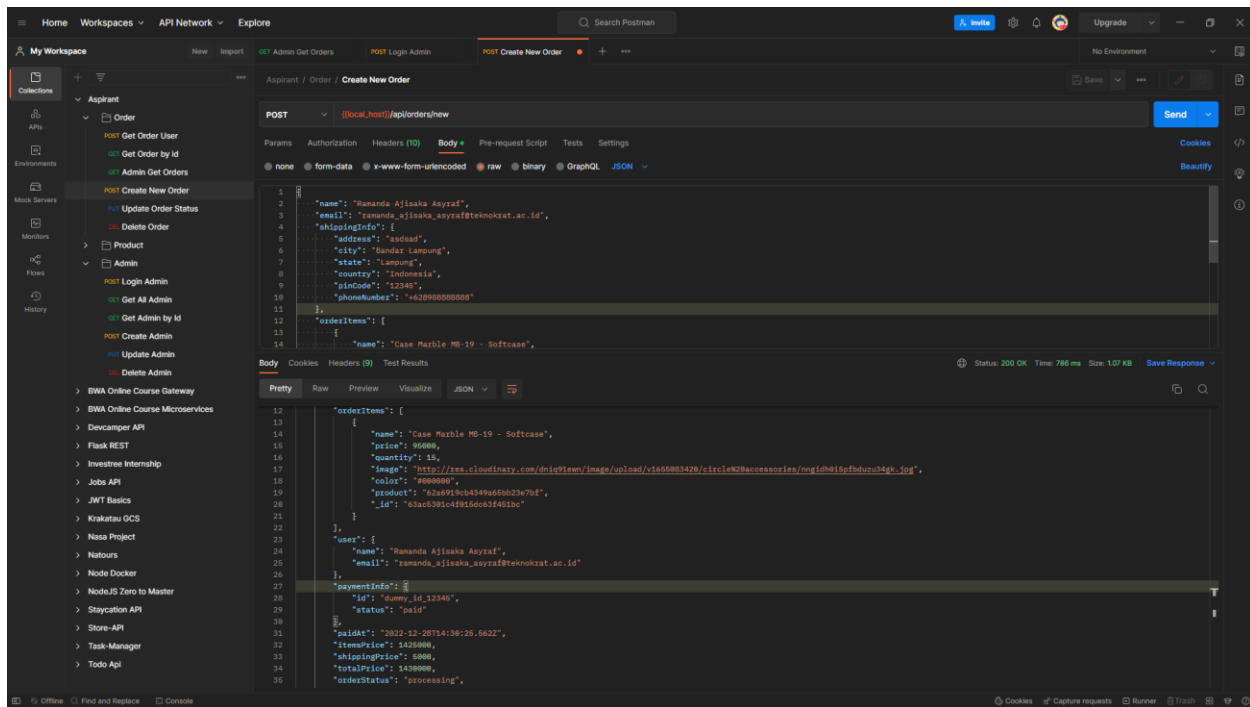


The screenshot shows a Postman workspace with a collection named 'Aspirant'. Inside the 'Order' folder, there is a GET request 'Admin Get Orders' with the URL '[[local\_host]]/api/admin/orders'. The request is sent, and the response is displayed in the 'Body' tab. The response is a JSON object with the following structure:

```
{
  "name": "Case Rasble MB-19 - Softcase",
  "price": 95000,
  "quantity": 1,
  "image": "http://res.cloudinary.com/dnig9iemw/image/upload/v1655883439/circleK20accessories/mgldh181pfdvuz4gk.jpg",
  "color": "#000000",
  "product": "62a6f0bb2d1be065409f07",
  "id": "62a6f0bb2d1be065409f07"
},
{
  "name": "CUSTOM CASE CUSTOM COSTUM CASE CASING CASE HP CUSTOM - Hardcase Full",
  "price": 35000,
  "quantity": 3,
  "image": "http://res.cloudinary.com/dnig9iemw/image/upload/v1655883558/circleK20accessories/muzcaasaoanobchpaw9v.jpg",
  "color": "#000000",
  "product": "62a6f0bb2d1be065409f07",
  "id": "62a6f0bb2d1be065409f07"
},
{
  "name": "Case Rasble MB-19 - Softcase",
  "price": 95000,
  "quantity": 1,
  "image": "http://res.cloudinary.com/dnig9iemw/image/upload/v1655883439/circleK20accessories/mgldh181pfdvuz4gk.jpg",
  "color": "#000000",
  "product": "62a6f0bb2d1be065409f07",
  "id": "62a6f0bb2d1be065409f07"
}
],
  "paidAt": "2022-06-25T01:42:39.615Z",
  "itemPrice": 345000,
  "shippingPrice": 10000,
  "totalPrice": 355000,
  "orderStatus": "confirmed"
}
```

Endpoint api ini digunakan untuk mendapatkan seluruh data orderan user. Pada saat melakukan request ke endpoint ini client harus mengirimkan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah GET.

## Create New Order (host\_server/api/orders/new)



Endpoint api ini digunakan untuk membuat sebuah orderan baru. Pada saat melakukan request ke endpoint ini client harus mengirimkan body sebagai berikut dengan method POST.

```
{
  "name": "string",
  "email": " string",
  "shippingInfo": {
    "address": " string",
    "city": " string",
    "state": " string",
    "country": " string",
    "pinCode": " string",
    "phoneNumber": " string"
  },
  "orderItems": [
    {
      "name": " string",
      "price": number,

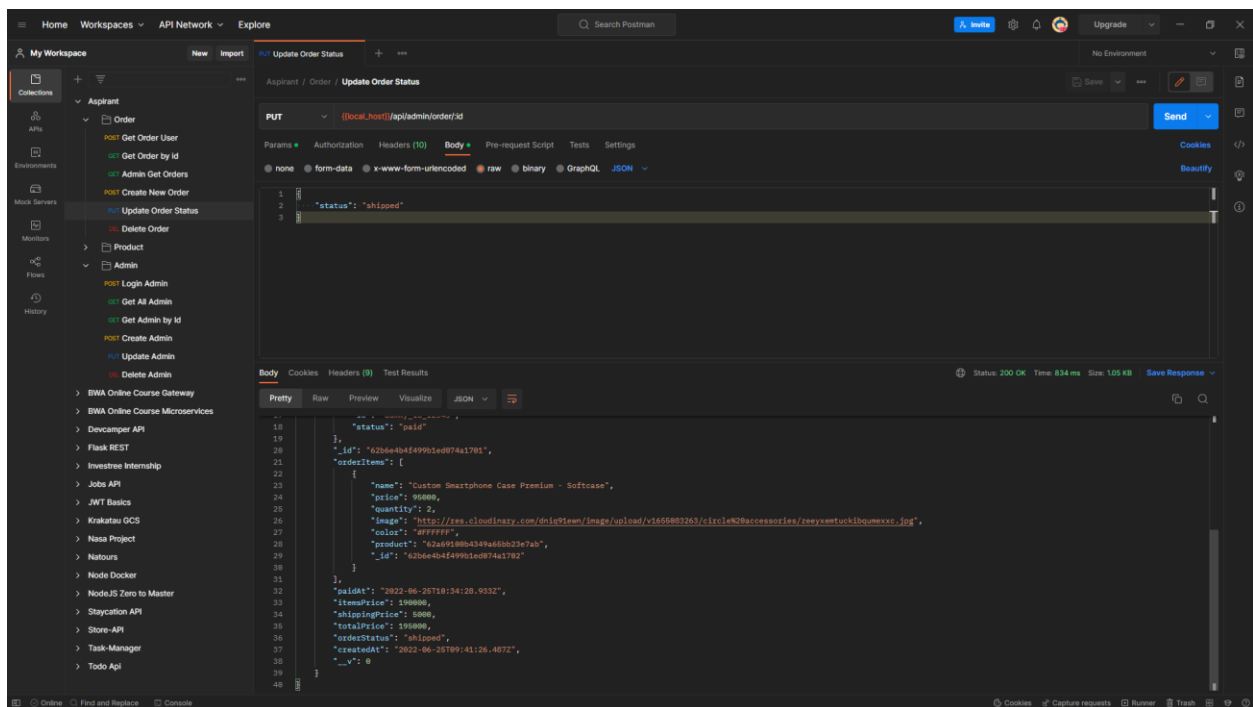
```

```

    "quantity": number,
    "image": "string",
    "color": "string",
    "product": "string"
  }
},
"paymentInfo": {
  "id": "string",
  "status": "paid"
},
"itemsPrice": number,
"shippingPrice": number,
"totalPrice": number
}

```

### Update Order Status (host\_server/api/admin/order/:id)



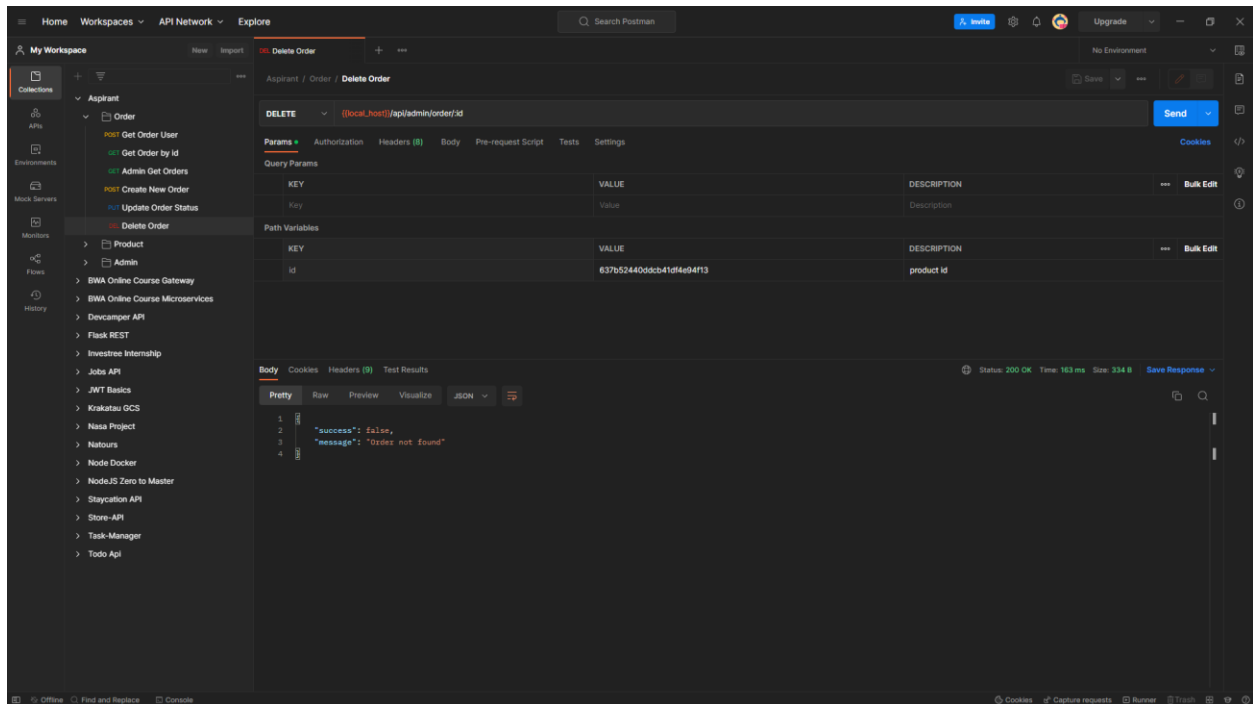
Endpoint api ini digunakan untuk mengupdate status orderan user. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variables id orderan, dan juga Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah PUT. Untuk bodynya adalah sebagai berikut:

```

{
  "status": "string"
}

```

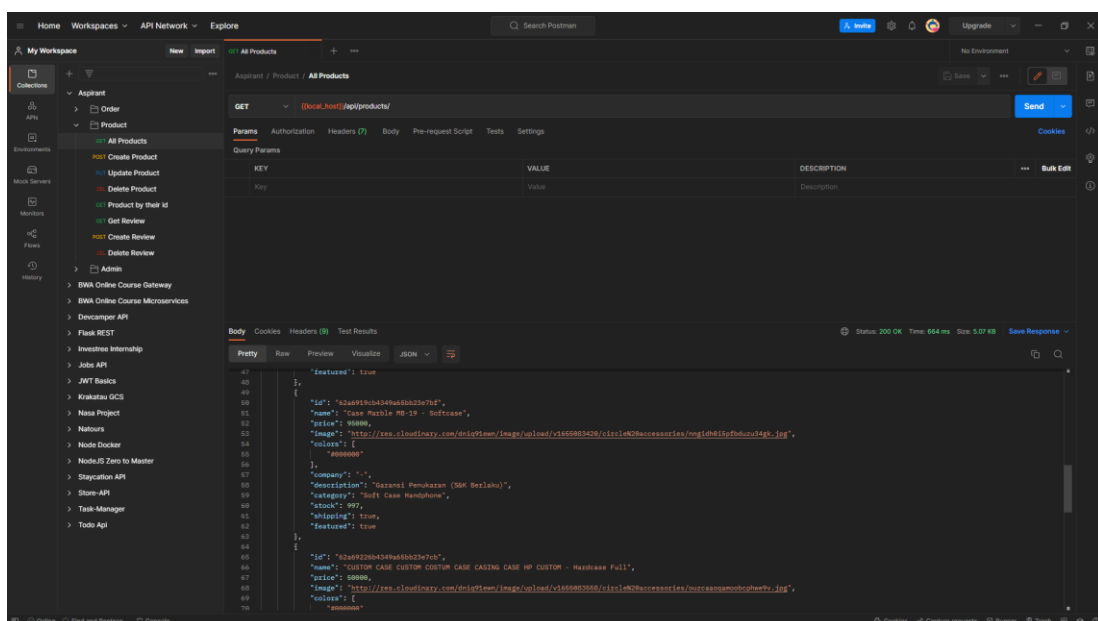
## Delete Order (host\_server/api/admin/order/:id)



Endpoint api ini digunakan untuk menghapus orderan user. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variables id orderan, dan juga Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah DELETE.

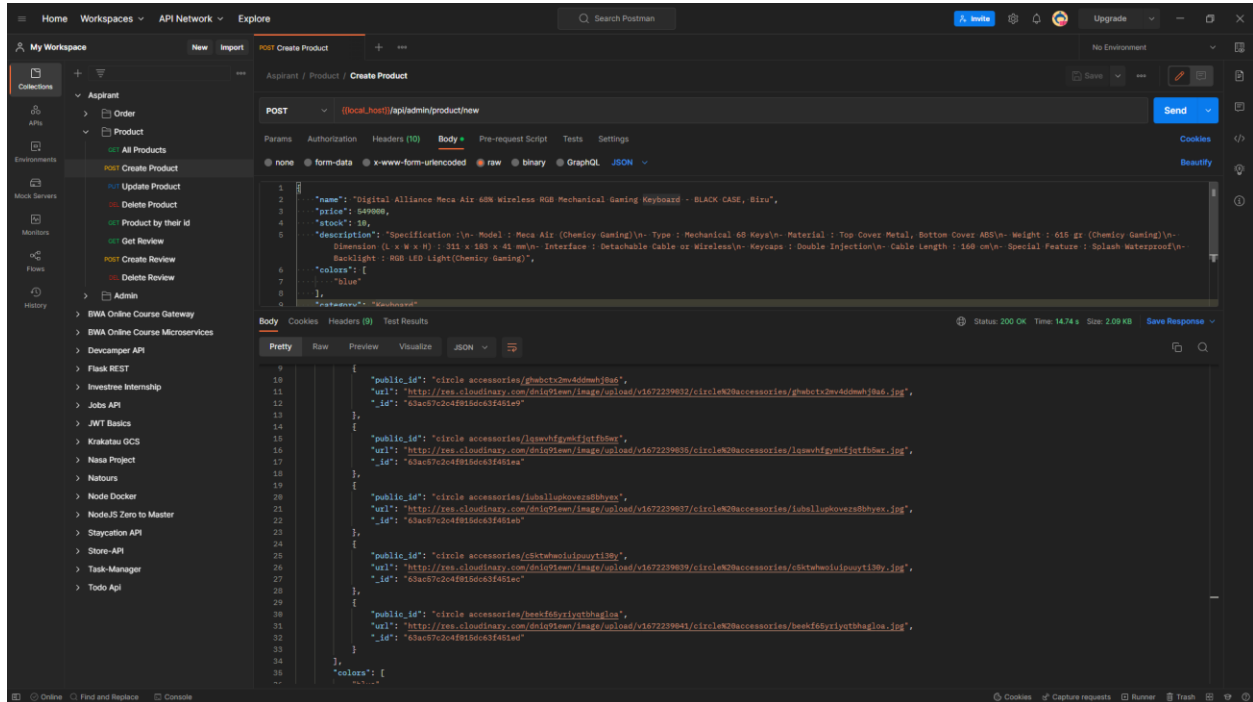
## B. Product Route

### Get All Product(host\_server/api/products/)



Endpoint api ini digunakan untuk mendapatkan seluruh product. Method yang digunakan adalah GET.

## Create Product (host\_server/api/admin/product/new)

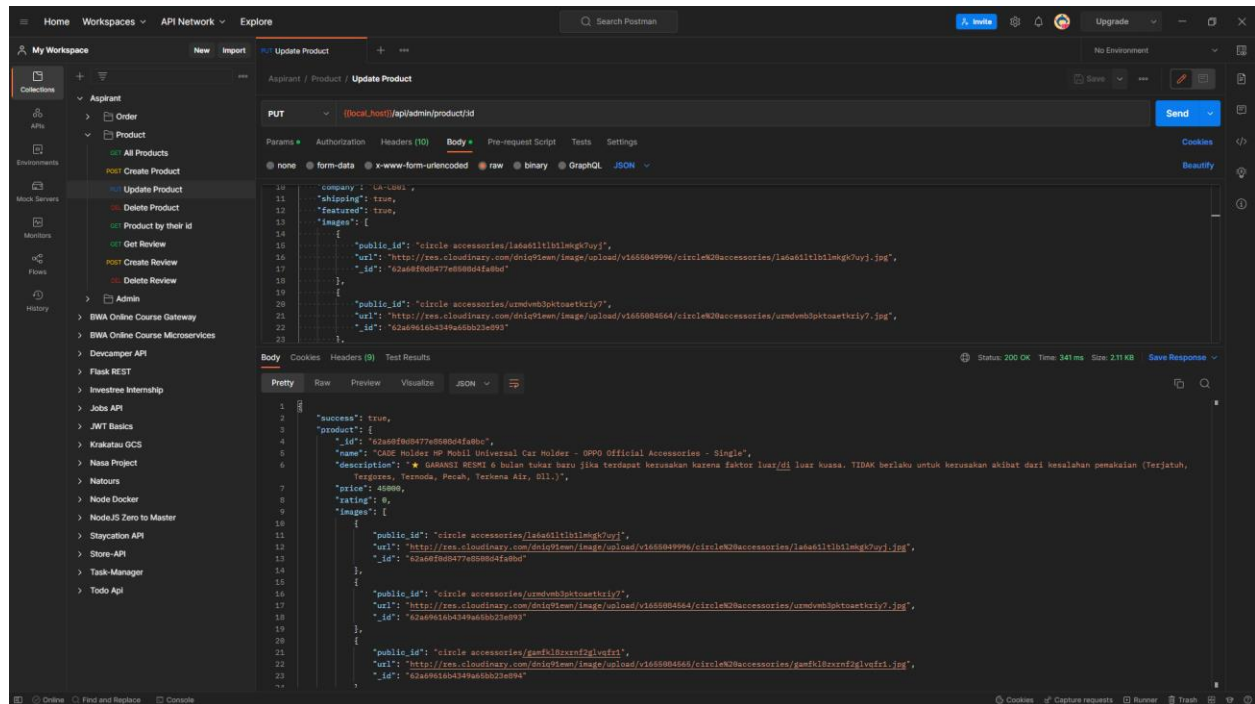


Endpoint api ini digunakan untuk membuat sebuah product baru. Pada saat melakukan request ke endpoint ini client harus mengirimkan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah POST. Untuk body yang akan dikirim sebagai berikut:

```
{  "name": "string",  "price": number,  "stock": number,  "description": "text",  "colors": [    "string"  ],  "category": "string",  "company": "string",  "shipping": boolean,  "featured": boolean,  "images": [ {} ]}
```



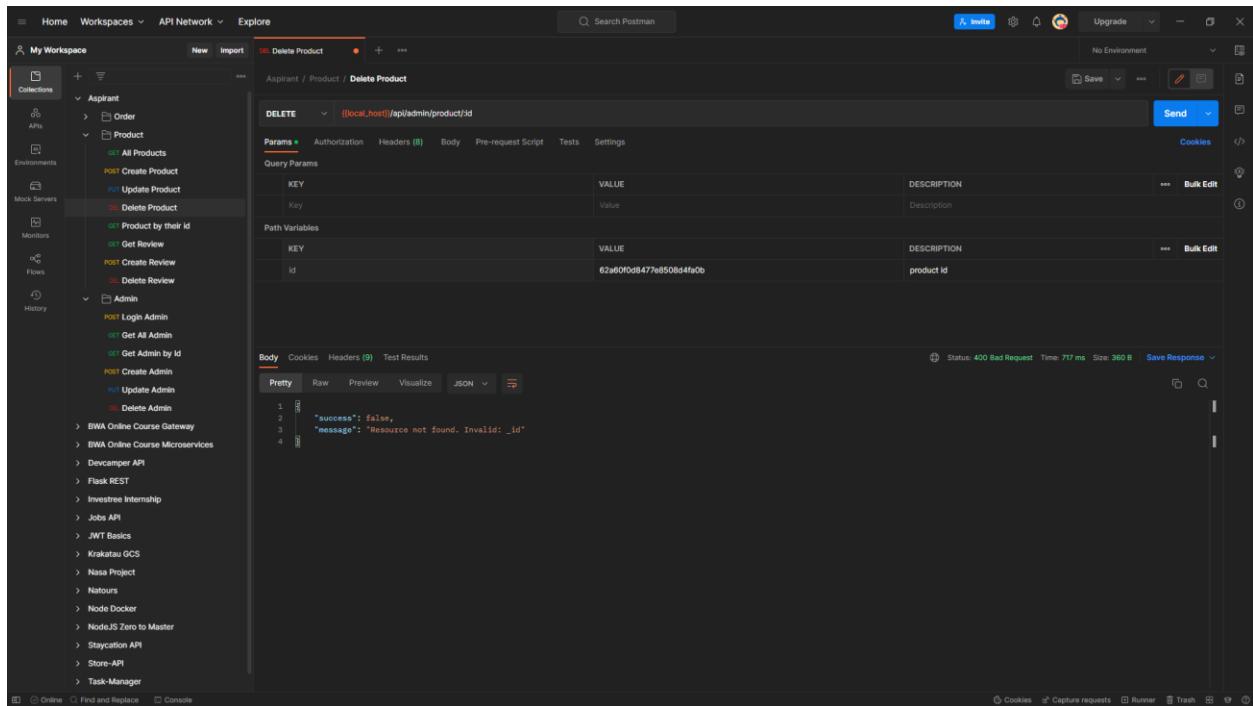
## Update Product (host\_server/api/admin/product/:id)



Endpoint api ini digunakan untuk mengupdate sebuah product. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id product Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah PUT. Untuk body yang akan dikirim sebagai berikut:

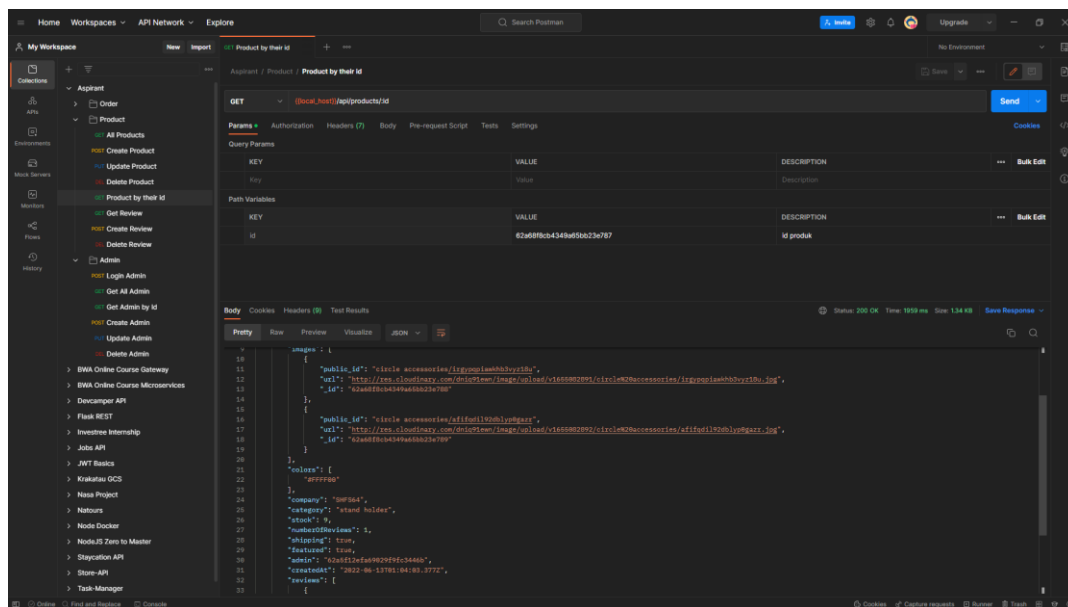
```
{
  "name": "string",
  "price": number,
  "stock": number,
  "description": "text",
  "colors": [
    "string"
  ],
  "category": "string",
  "company": "string",
  "shipping": boolean,
  "featured": boolean,
  "images": [ {} ]
}
```

### Delete Product (host\_server/api/admin/product/:id)



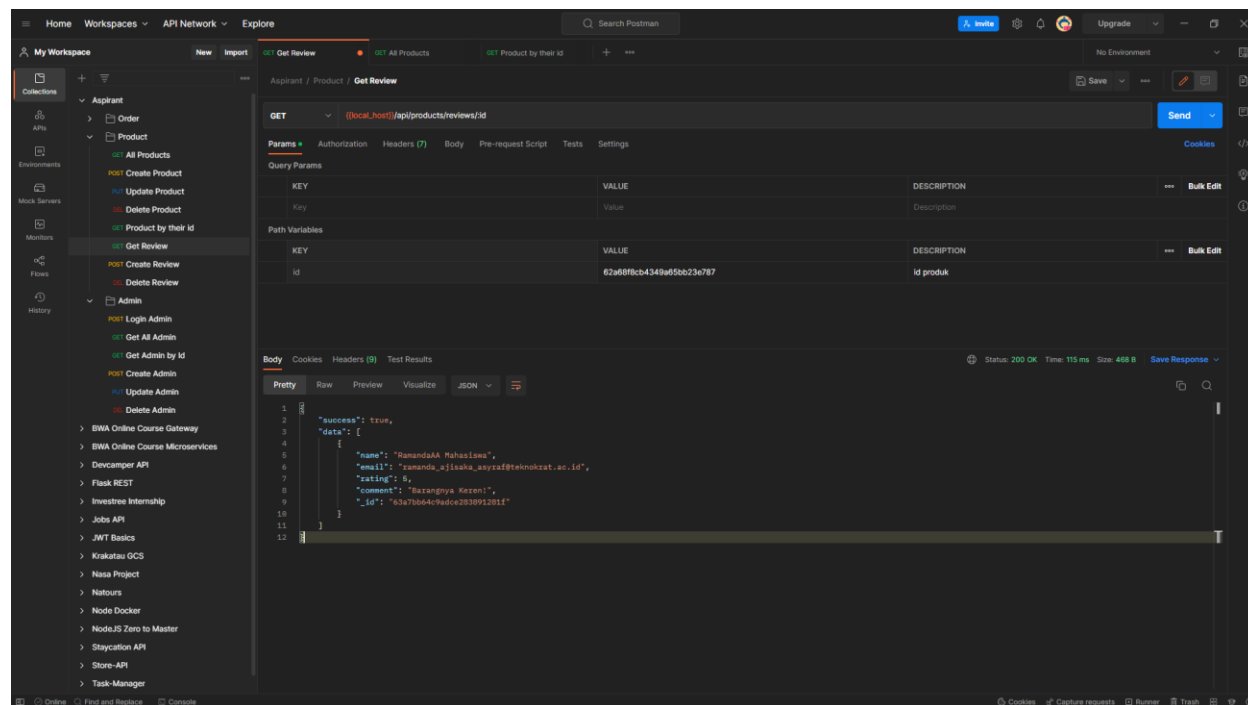
Endpoint api ini digunakan untuk menghapus sebuah product. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id product Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah DELETE.

### Get Product by Id (host\_server/api/products/:id)



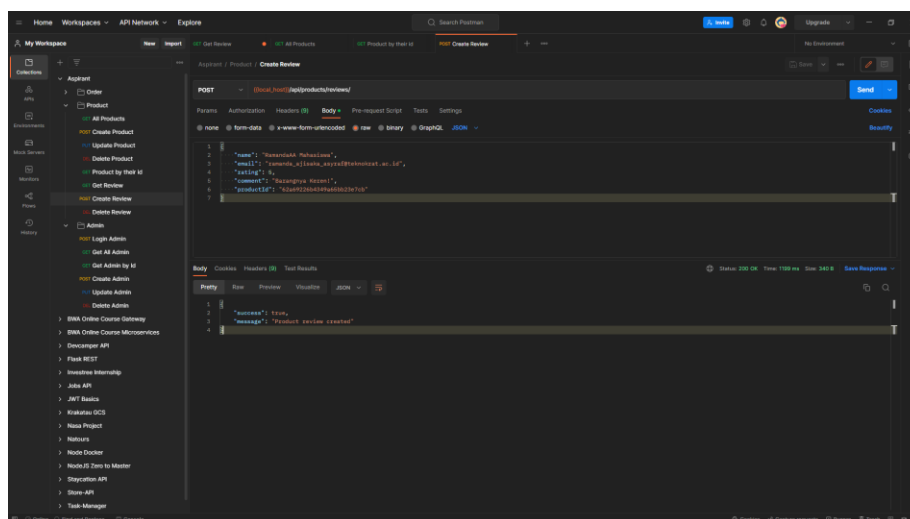
Endpoint api ini digunakan untuk mengambil data sebuah product berdasarkan idnya. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id product Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah GET.

### Get Review Product (host\_server/api/products/reviews/:id)



Endpoint api ini digunakan untuk mengambil data review sebuah product berdasarkan idnya. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id product, Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah GET.

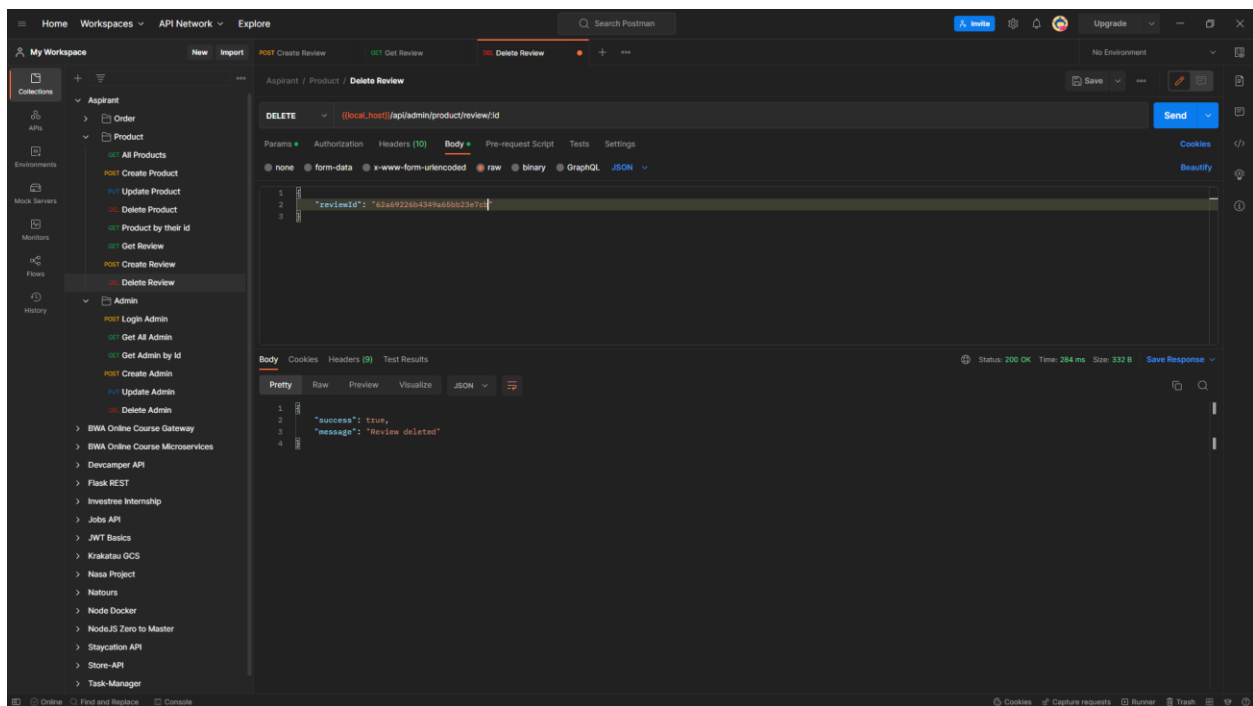
### Create Review (host\_server/api/products/reviews/)



Endpoint api ini digunakan untuk membuat data review sebuah product, lalu method yang digunakan adalah POST. Untuk body yang dikirimkan adalah sebagai berikut:

```
{
  "name": "string",
  "email": "string",
  "rating": number,
  "comment": "string",
  "productId": "string"
}
```

### Delete Review (host\_server/api/admin/product/review/:id)

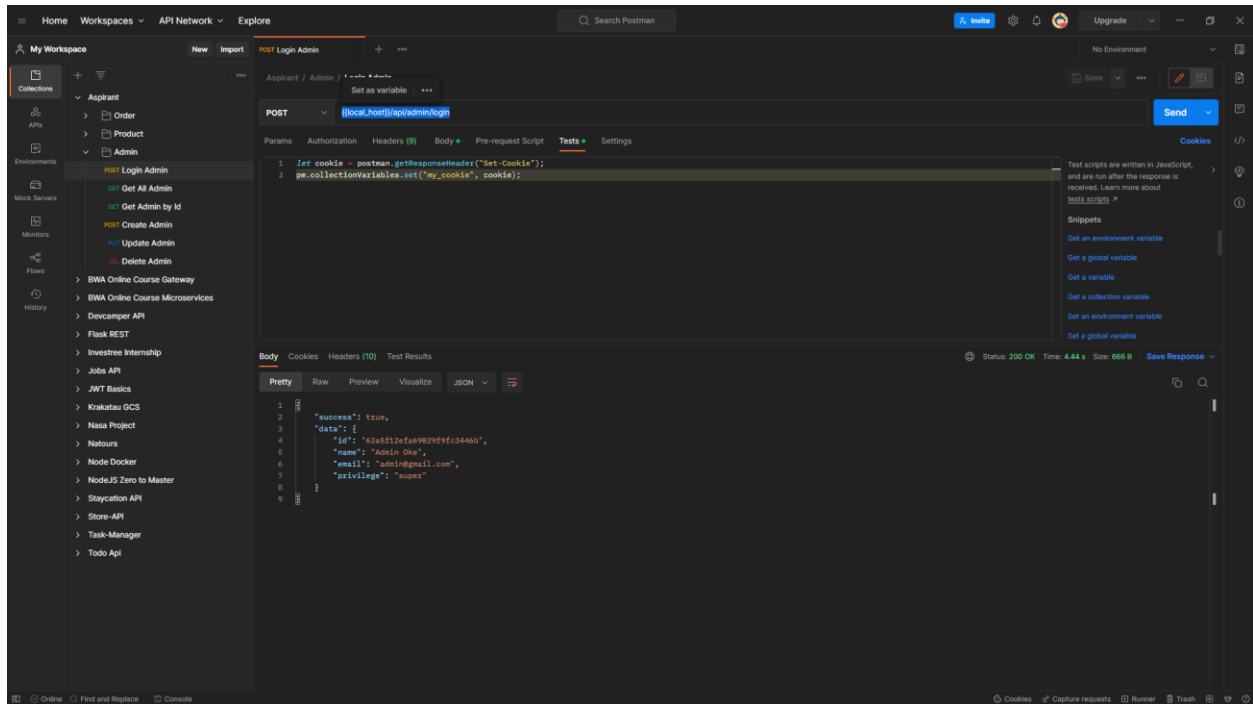


Endpoint api ini digunakan untuk menghapus data review sebuah product berdasarkan idnya. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id product, Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah DELETE. Untuk bodynya sebagai berikut:

```
{
  "reviewId": "string"
}
```

## C. Admin Route

### Login Admin (host\_server/api/admin/login)

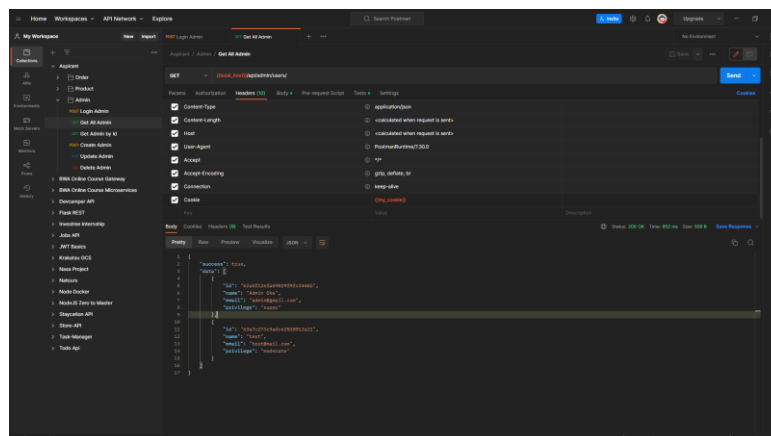


Endpoint api ini digunakan untuk admin login, lalu method yang digunakan adalah POST.

Untuk bodynya sebagai berikut:

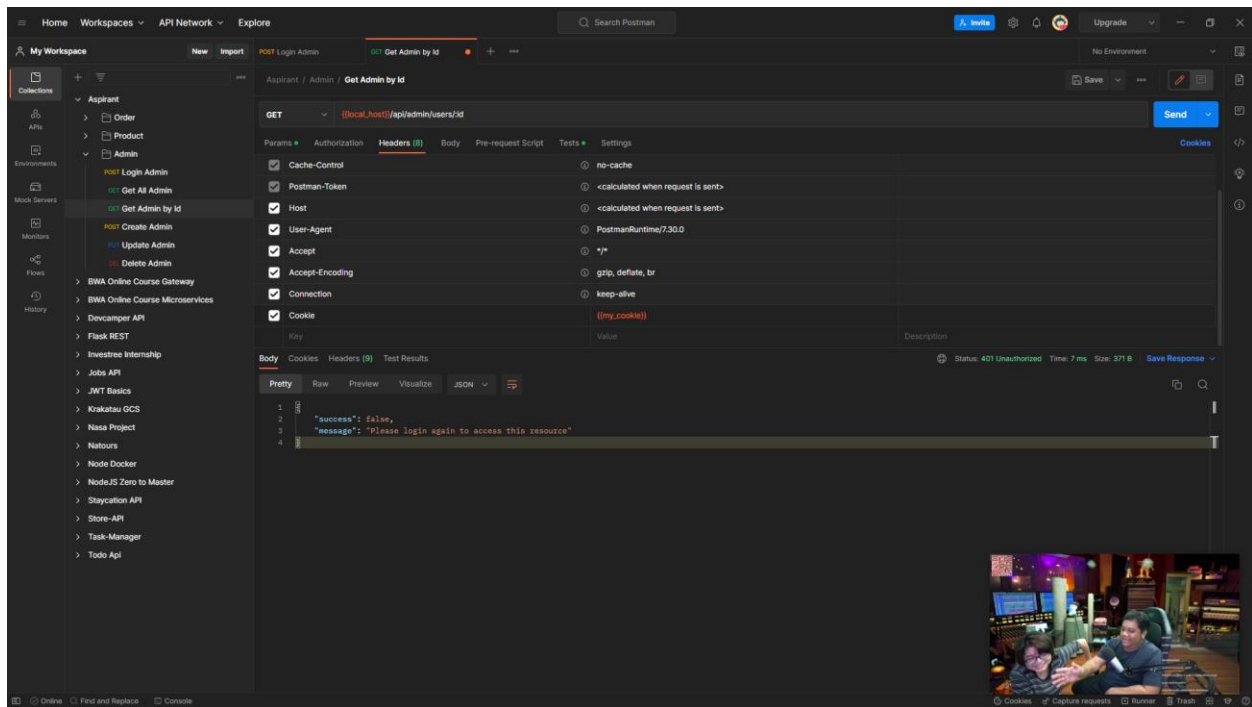
```
{
  "email": "string",
  "password": "string"
}
```

### Get All Admin (host\_server/api/admin/users/)



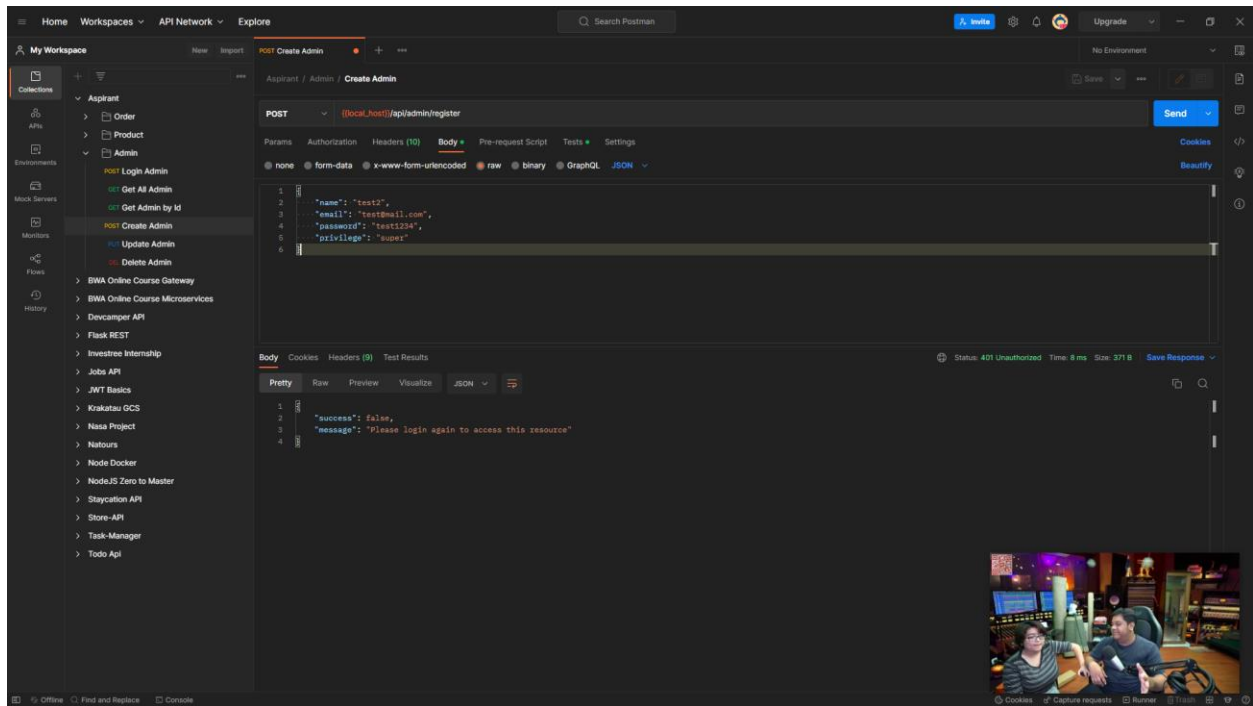
Endpoint api ini digunakan untuk mengambil data seluruh admin. Pada saat melakukan request ke endpoint ini client harus mengirimkan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah GET.

### Get Admin by Id (host\_server/api/admin/users/:id)



Endpoint api ini digunakan untuk mengambil data admin berdasarkan idnya. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id ,dan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah GET.

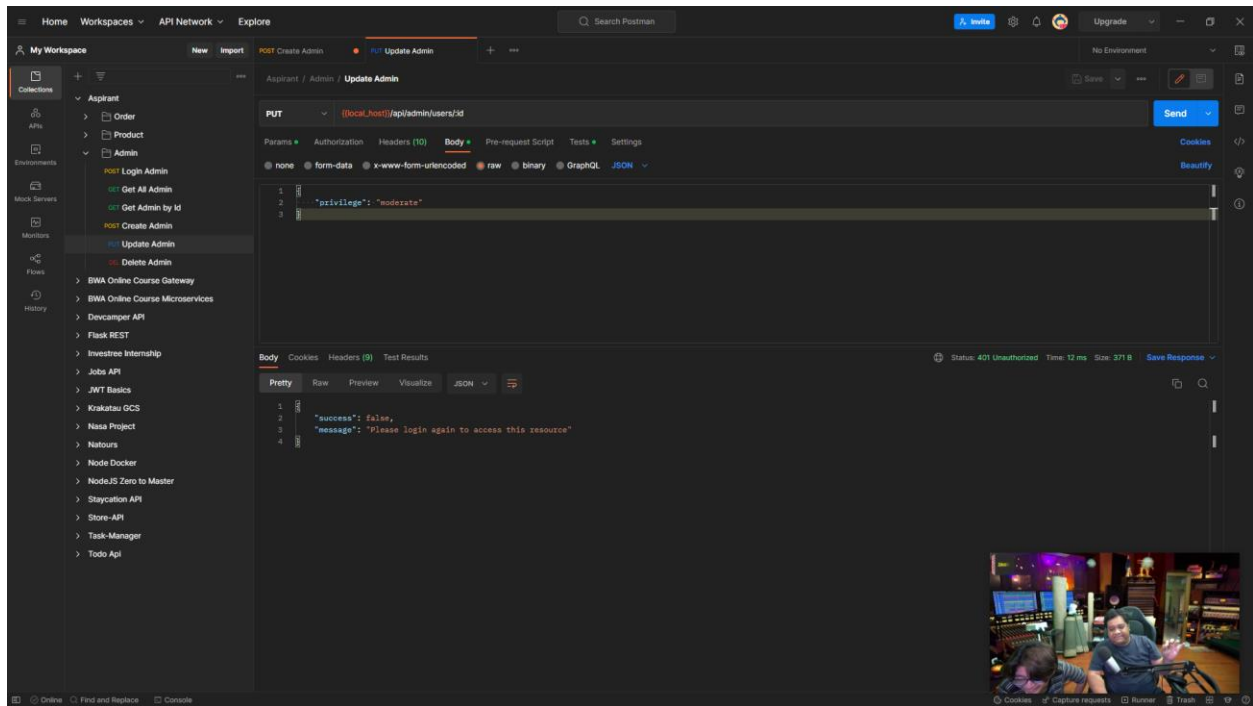
## Create Admin (host\_server/api/admin/register)



Endpoint api ini digunakan untuk membuat admin baru. Pada saat melakukan request ke endpoint ini client harus mengirimkan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah POST. Body yang dikirimkan adalah sebagai berikut:

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "privilege": "string"
}
```

## Update Admin (host\_server/api/admin/users/:id)

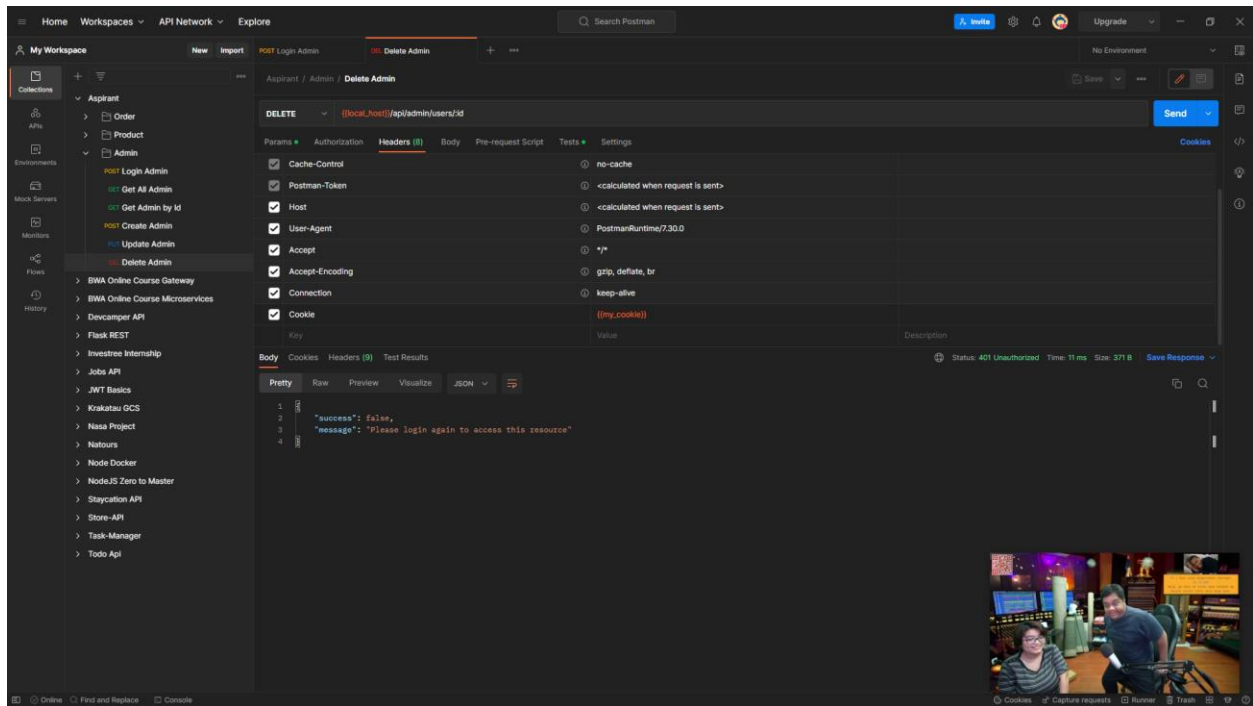


Endpoint api ini digunakan untuk membuat admin baru. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id admin, dan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah PUT. Body yang dikirimkan adalah sebagai berikut:

```
{  "privilege": "string"}
```



## Delete Admin (host\_server/api/admin/users/:id)



Endpoint api ini digunakan untuk menghapus admi. Pada saat melakukan request ke endpoint ini client harus mengirimkan path variable id admin, dan Headers Cookie berisikan token yang diberikan pada saat login sebagai admin, lalu method yang digunakan adalah DELETE.