1. Register

Route : http://localhost/dress\_marketplace/api/register

Method : POST

Body :

* email : text
* password : text
* full\_name : text
* gender : ‘M’ atau ‘F’
* phone\_number : text

Kemungkinan output:

* 'status'=>false,'message'=>'Email already exist'
* 'status'=>true,'message'=>'User created successfully'

1. Login

Route : http://localhost/dress\_marketplace/api/login

Method : POST

Body :

* email : text
* password : text

Kemungkinan output:

* 'status'=>false,'message'=>'User Not Found'
* 'status'=>false,'message'=>'User Non-Active'
* 'status'=>false,'message'=>'Invalid Credentials'
* 'status'=>true,'jwt'=>xxxxxxxxxxxxxxxxxxx

1. Get Auth User

Route : http://localhost/dress\_marketplace/api/get\_auth\_user

Method : POST

Body :

* token : text

output:

* 'error'=>’blabla’ (statuscode 500)
* 'status'=>true, 'result'=>blabla

**Jalankan 3 perintah ini dulu :**

composer install

php artisan migrate

php artisan db:seed

1. Check Store Name

Route : http://localhost/dress\_marketplace/api/check\_store\_name

Method : POST

Body :

* store\_name : text

output:

* ‘status’ => true, ‘message’ => ‘Store Name Available’
* ‘status’ => false, ‘message’ => ‘Store Name Already Exists’

1. Register Store Name

Route : http://localhost/dress\_marketplace/api/register\_store\_name

Method : POST

Body :

* token : text
* store\_name : text

output:

* ‘status’ => false, ‘message’ => ‘Store Name Already Exists’
* ‘status’ => false, ‘message’ => ‘You already have store’
* 'error'=>'something went wrong, try again later'
* ‘status’ => true, ‘message’ => ‘Store Name registered Successfully’

1. Get Province List

Route : http://localhost/dress\_marketplace/api/get\_province\_list

Method : POST

Body :

output:

* ‘province’ => data

1. Get City by Province

Route : http://localhost/dress\_marketplace/api/get\_city\_by\_province

Method : POST

Body :

* province\_id : text

output:

* ‘city’ => data

1. Get Courier List

Route : http://localhost/dress\_marketplace/api/get\_courier\_list

Method : POST

Body :

output:

* ‘courier’ => data

1. Register Store

Route : http://localhost/dress\_marketplace/api//register\_store

Method : POST

Body :

* token : text
* store\_name : text
* photo : file (boleh null)
* banner : file (boleh null)
* description : text
* established\_year : text
* province : text => berupa id yang didapat dari get province diatas
* city : text => berupa id yang didapat dari get city diatas
* business\_type : text
* contact\_person\_name : text (boleh null)
* contact\_person\_job\_title : text (boleh null)
* contact\_person\_phone\_number : text (boleh null)
* courier : text => berupa array berisi id yang didapat dari get courier list
* ktp : file (boleh null)
* siup : file (boleh null)
* npwp : file (boleh null)
* skdp : file (boleh null)
* tdp : file (boleh null)

output:

* ‘status’ => false, ‘message’ => ‘You don’t have privilege’
* ‘status’ => false, ‘message’ => ‘error : blabla’
* ‘status’ => true, ‘message’ => ‘Store registered successfully’

Jalankan perintah ini:

php artisan migrate

php artisan db:seed --class=DressAttributeSeeder

1. Get Dress Attributes

Route : http://localhost/dress\_marketplace/api/get\_dress\_attributes

Method : POST

Body :

output:

* 'style' : [],
* 'price' : [],
* 'size' : [],
* 'season' : [],
* 'neckline' : [],
* 'sleevelength' : [],
* 'waiseline': [],
* 'material': [],
* 'fabrictype': [],
* 'decoration': [],
* 'patterntype': []

Yang login tambah 1 response :

* 'status'=>false,'message'=>'User Non-Active'

1. Get User Store

Route : http://localhost/dress\_marketplace/api/get\_user\_store

Method : POST

Body :

* token

output:

* ‘have\_store’=> false, ‘store’ => null
* ‘have\_store’=> true, ‘store’ => ~~~

Yang Register Store sudah direvisi, couriernya menjadi multiple

* courier : text => berupa array berisi id yang didapat dari get courier list

**contoh :**

courier[0] : 1

courier[1] : 2

1. Add Product

Route : http://localhost/dress\_marketplace/api/add\_product

Method : POST

Body :

* Token : text
* Name : text (nama barang)
* Min\_order : text
* Weight : text (dalam satuan gram)
* Description : text
* Style\_id : text
* Season\_id : text
* neckline\_id : text
* sleevelength\_id : text
* waiseline\_id : text
* material\_id : text
* fabrictype\_id : text
* decoration\_id : text
* patterntype\_id : text
* size[] : text -> berupa array yang berisi id size
* price[] : text-> berupa array yang berisi informasi qty\_min, qty\_max, price
* photo : file

Yang price contohnya gini :

0 – 10 harganya 100000

11-20 harganya 90000

>21 harganya 80000

Jadi tulisnya gini :

Price[0][qty\_min] = 0

Price[0][qty\_max] = 10

Price[0][qty\_price] = 100000

Price[1][qty\_min] = 11

Price[1][qty\_max] = 20

Price[1][qty\_price] = 90000

Price[2][qty\_min] = 21

Price[2][qty\_max] = max

Price[2][qty\_price] = 80000

**Contoh**

****

output:

* ‘status=> false, ‘message’ => ‘You don’t have privilege’
* ‘status=> false, ‘message’ => ‘error’
* ‘status=> true, ‘message’ => ‘Product Registered Successfully’

Kalau mau lebih jelas coba liat yang versi Web : <http://localhost/dress_marketplace/seller_panel_product>

Setelah ekstrak folder Courier, jalankan :

**php artisan db:seed --class=CourierSeeder**

1. Get Product Detail

Route : http://localhost/dress\_marketplace/api/get\_product\_detail

Method : POST

Body :

* product\_id : text

output:

* ‘status’ : false, message : ‘product nonActive’
* ‘status’ : false, message : ‘product doesn’t exist’
* ‘status’ : true, product\_info, store\_info

**php artisan migrate**

1. Add to Bag

Route : http://localhost/dress\_marketplace/api/add\_to\_bag

Method : POST

Body :

* token : text
* product\_id : text
* product\_size\_qty : array of object

output:

* ‘status’ : false, message : error
* ‘status’ : true, message : ‘Product Added to Cart Successfully’

Contoh :



1. View Shopping Bag

Route : http://localhost/dress\_marketplace/api/view\_shopping\_bag

Method : POST

Body :

* token : text

output:

* ‘status’ : false, message : error
* ‘status’ : true, bag, total\_qty, total\_price

Hasilnya adalah grouping per toko, kemudian per produk, kemudian baru rincian ke size



1. Delete Product from Bag

Route : http://localhost/dress\_marketplace/api/delete\_product\_from\_bag

Method : POST

Body :

* token : text
* product\_id : text

output:

* ‘status’ : false, ‘message’ : error
* ‘status’ : true, ‘message’ : ‘<Nama Produk> Deleted from Bag’

Setelah ekstrak folder Bank, jalankan :

**php artisan db:seed --class=BankSeeder**

**php artisan migrate**

1. Get Checkout Info => harus connect internet, karena ambil API dari third-party

Route : http://localhost/dress\_marketplace/api/ get\_checkout\_info

Method : POST

Body :

* token : text
* destination\_city : text (berupa id kota tujuan / kota pembeli)

Output:

* informasi per toko, masing-masing toko ada beli produk apa saja, kemudian courier yang tersedia sesuai toko tersebut dari kota tokonya ke kota tujuan (destination\_city)
* total\_qty, total\_price, available\_points

1. Checkout

Route : http://localhost/dress\_marketplace/api/checkout

Method : POST

Body :

* token : text
* receiver\_name : text
* address : text
* province: text (berupa id angka)
* city : text (berupa id angka)
* phone\_number : text
* postal\_code : text
* use\_point : text (validasi tidak boleh melebihi balance)
* courier : berupa array of object (store\_id, courier\_id, courier\_service, fee, note)



output:

* ‘status’ : false, ‘message’ : error
* ‘status’ : true, ‘message’ : ‘Checkout Successfully’, ‘data’ : transaction\_id, total\_price, bank

1. Get Purchase Payment

Fungsi : User untuk tau dia perlu bayar invoice apa saja, terdiri dari 2 status : Waiting for Payment dan Payment Confirmation Sent. Setelah diapprove admin baru hilang.

Route : http://localhost/dress\_marketplace/api/get\_purchase\_payment

Method : POST

Body :

* token : text

Output :

* result : daftar invoice
* bank : master bank, yg bisa dipilih saat konfirmasi pembayaran

1. Confirm Payment

Route : http://localhost/dress\_marketplace/api/confirm\_payment

Method : POST

Body :

* token : text
* transaction\_id : text
* company\_bank\_id : text
* amount : text
* sender\_bank : text
* sender\_account\_number : text
* sender\_name : text
* note : text

Output :

* success : true, message : Invoice (transaction\_id) Payment Confirmation Submitted Successfully



Berikutnya admin approve melalui web : <http://localhost/dress_marketplace/admin/verify_payment>

1. Seller Get Order

Fungsi : Setelah pembayaran diapprove admin, maka order dapat dilihat di sini melalui seller panel. Berisi daftar order dari pembeli

Route : http://localhost/dress\_marketplace/api/seller\_get\_order

Method : POST

Body :

* token : text

Output :

* status: true, result : []



1. Get Order Status

Fungsi : dari sisi User untuk melihat orderannya. Terdiri dari 2 jenis order\_status : ‘Waiting Seller Response’ dan ‘Order Approved’. Jika ‘Order Approved’, maka muncul rincian mana yg accept mana yg reject.

Order ini akan tetap ada, sampai diterima barangnya baru hilang.

Route : http://localhost/dress\_marketplace/api/get\_order\_status

Method : POST

Body :

* token : text

Output :

* status: true, result : []



1. Approve Order Product

Fungsi : Penjual untuk response pesanan, mana product yang accept, mana yg reject. Accept pake angka 1, reject pake angka 2.

Route : http://localhost/dress\_marketplace/api/approve\_order\_product

Method : POST

Body :

* token : text
* transaction\_id : text
* store\_id : text
* product : array of object

Output :

* status: true, message : Approved Successfully



Kemudian, kalau dari user get\_order\_status lagi, maka status sudah berubah menjadi Order Approved :



Dan ditampilkan mana yang accept, mana yg reject:



1. Seller Get Shipping Confirmation

Fungsi : berisi daftar pesanan yang sudah diapprove sebelumnya kemudian siap dikirim.

Route : http://localhost/dress\_marketplace/api/seller\_get\_shipping\_confirmation

Method : POST

Body :

* token : text

Output :

* status : true, result : []



Produk yang ditampilkan hanya yang diaccept sebelumnya:



Namun, jika direject semua, artinya gak ada produk yang perlu dikirim, maka munculnya No Product



Jadi, penyelesaiannya ada 2 opsi:

1. Jika ada produk, maka input nomor resi seperti normal (nomor 25 dibawah), nanti muncul di tempat pembeli informasi pengirimannya
2. Jika No Product, maka Finish Shipping (nomor 26 dibawah), langsung done, uang produknya lgsung kembali ke pembeli, tapi ongkos kirimnya masuk ke penjual :v

Jadi nanti tolong divalidasi, jika product == ‘No Product’ maka tampilkan tombol Finish Shipping.

Jika product ada produk, maka tampilkan inputan nomor resi

1. Input Receipt Number

Fungsi : Penjual menginput nomor resi pengiriman

Route : http://localhost/dress\_marketplace/api/input\_receipt\_number

Method : POST

Body :

* token : text
* transaction\_id : text
* store\_id : text
* receipt\_number : text

Output :

* status : true, message : ‘Submitted Successfully’



1. Finish Shipping

Fungsi : Langsung done order, karena gak ada produk yang dikirim

Route : http://localhost/dress\_marketplace/api/finish\_shipping

Body :

* token : text
* transaction\_id : text
* store\_id : text

Output :

* status : true, message : ‘Submitted Successfully’



1. Get Receipt Confirmation

Fungsi : Pembeli untuk melihat pengiriman orderannya, berupa nomor resi

Route : http://localhost/dress\_marketplace/api/get\_receipt\_confirmation

Body :

* token : text

Output :

* status : true, result : []



1. Confirm Receipt

Fungsi : Pembeli untuk mengkonfirmasi bahwa dia sudah menerima barang

Route : http://localhost/dress\_marketplace/api/confirm\_receipt

Body :

* token : text
* transaction\_id : text
* store\_id : text

Output :

* status : true, message : ‘Confirm Successfully’



Setelah konfirmasi ini, maka :

1. barang yang direject, nominalnya kembali ke pembeli
2. barang yang accept + ongkos kirim, nominalnya ke penjual

Jadi sama dengan yang finish shipping tadi :

1. barang yang direject, kembali ke pembeli
2. ongkos kirim, nominalnya ke penjual
3. Add to Wishlist

Route : http://localhost/dress\_marketplace/api/add\_to\_wishlist

Body :

* token : text
* product\_id : text

Output :

* status : true , message : Product added to Wishlist Successfully

1. Delete from Wishlist

Route : http://localhost/dress\_marketplace/api/delete\_from\_wishlist

Body :

* token : text
* product\_id : text

Output :

* status : true , message : Product deleted from wishlist

yang Get Product Detail ada sedikit penambahan, karena sudah ada wishlist.

Get Product Detail

Route : http://localhost/dress\_marketplace/api/get\_product\_detail

Method : POST

Body :

* token : text -> opsional
* product\_id : text

output:

* ‘status’ : false, message : ‘product nonActive’
* ‘status’ : false, message : ‘product doesn’t exist’
* ‘status’ : true, product\_info, store\_info, wishlist\_status

Pada outputnya ada tambah 1 informasi : wishlist\_status bernilai true atau false.

Token bersifat opsional, karena tidak login pun bisa liat product detail.

1. My Wishlist

Route : http://localhost/dress\_marketplace/api/my\_wishlist

Body :

* token : text

Output :

* status : true , result : []

1. Withdraw

Route : http://localhost/dress\_marketplace/api/withdraw

Body :

* token : text
* amount : text
* bank\_name : text
* branch : text
* account\_number : text
* name\_in\_account : text
* password : text

Output :

* status : false, message : Insufficient balance
* status : false, message : Wrong Password
* status : true, message : Submitted Successfully

1. Get Review Rating

Route : http://localhost/dress\_marketplace/api/get\_review\_rating

Body :

* token : text

Output :

* status : true, result : []

1. Submit Review Rating

Route : http://localhost/dress\_marketplace/api/submit\_review\_rating

Body :

* token : text
* transaction\_id : text
* store\_id : text
* store\_rating : text
* product\_rating : array of object

Output :

* status : true, message : ‘Submitted Successfully’



1. Transaction History

Route : http://localhost/dress\_marketplace/api/transaction\_history

Body :

* token : text

Output :

* status : true, result : []

1. Get Request Partnership

Route : http://localhost/dress\_marketplace/api/get\_request\_partnership

Body :

* token : text

Output :

* status : true, result : []

Pada masing-masing produk terdapat field has\_partnership yang bernilai true / false. Jika false, maka belum memiliki partnership dgn produk tsb, jadi bisa request. Sedangkan jika true, maka partnership sudah diajukan (Waiting Approval). Partnership yang sudah diaccept / reject nantinya gak akan muncul di Get Request Partnership ini lagi.

1. Submit Request Partnership

Route : http://localhost/dress\_marketplace/api/submit\_request\_partnership

Body :

* token : text
* product\_id : text
* min\_order : text
* price[] : text-> berupa array yang berisi informasi qty\_min, qty\_max, price

Output :

* status : true, message : “Request Partnership Submitted Successfully”



1. Upline Get Request Partnership

Fungsi : Upline untuk melihat list request dari downline

Route : http://localhost/dress\_marketplace/api/upline\_get\_request\_partnership

Body :

* token : text

Output :

* status : true, result : []

1. Accept Partnership

Route : http://localhost/dress\_marketplace/api/accept\_partnership

Body :

* token : text
* partnership\_id : text

Output :

* status : true, message : ‘Partnership Accepted’

1. Reject Partnership

Route : http://localhost/dress\_marketplace/api/reject\_partnership

Body :

* token : text
* partnership\_id : text

Output :

* status : true, message : ‘Partnership Rejected’

1. Upline Partner List

Route : http://localhost/dress\_marketplace/api/upline\_partner\_list

Body :

* token : text

Output :

* status : true, result : []

1. Downline Partner List

Route : http://localhost/dress\_marketplace/api/downline\_partner\_list

Body :

* token : text

Output :

* status : true, result : []

1. Get Store Detail

Route : http://localhost/dress\_marketplace/api/get\_store\_detail

Body :

* store\_id : text

Output :

* status : true, result : []

Alternatif 2:

Get User Store Detail

Route : http://localhost/dress\_marketplace/api/get\_user\_store\_detail

Body :

* token : text
* store\_id : text

Output :

* status : true, result : []

1. Add RFQ Request

Route : http://localhost/dress\_marketplace/api/add\_rfq\_request

Body :

* token : text
* item\_name : text
* description : text
* qty : text
* request\_expired : text (dalam format yyyy-mm-dd hh:mm:ss)
* budget\_unit\_min : text
* budget\_unit\_max : text
* photo : file

Output :

* status : true, message : ‘Submitted Successfully’

1. Seller Get RFQ Request

Route : http://localhost/dress\_marketplace/api/seller\_get\_rfq\_request

Body :

* token : text

Output :

* status : true, result : []

1. Add RFQ Offer

Route : http://localhost/dress\_marketplace/api/add\_rfq\_offer

Body :

* token : text
* rfq\_request\_id : text
* description : text
* price\_unit : text
* weight\_unit : text
* photo : file

Output :

* status : true, message : ‘Submitted Successfully’



1. View Active RFQ Request

Route : http://localhost/dress\_marketplace/api/view\_active\_rfq\_request

Body :

* token : text

Output :

* status : true, result : []

1. Accept RFQ Offer

Route : http://localhost/dress\_marketplace/api/accept\_rfq\_offer

Body :

* token : text
* rfq\_offer\_id : text

Output :

* status : true, message : ‘Submitted Successfully’

1. Close RFQ Request

Route : http://localhost/dress\_marketplace/api/close\_rfq\_request

Body :

* token : text
* rfq\_request\_id : text

Output :

* status : true, message : ‘Submitted Successfully’

1. RFQ Request History

Route : http://localhost/dress\_marketplace/api/rfq\_request\_history

Body :

* token : text

Output :

* status : true, result : []

1. Financial History

Route : http://localhost/dress\_marketplace/api/financial\_history

Body :

* token : text
* year : text (dalam format 4 digit, contoh : 2018)
* month : text (dalam format 2 digit, contoh : 07)

Output :

* status : true, result : []

1. RFQ Offer History

Route : http://localhost/dress\_marketplace/api/rfq\_offer\_history

Body :

* token : text

Output :

* status : true, result : []

1. Get New Product Detail

Route : http://localhost/dress\_marketplace/api/get\_new\_product\_detail

Body : kosong

Output:

* Status:true, product\_info :[]

1. Best Seller Product Detail

Route : http://localhost/dress\_marketplace/api/best\_seller\_product\_detail

Body : kosong

Output:

* Status:true, product\_info :[]

1. Search

Route : http://localhost/dress\_marketplace/api/search

Body :

* product\_name : text

Output:

* Status:true, product\_info :[] ,query,count,count\_all



1. Advance Search (Filter)

Route : http://localhost/dress\_marketplace/api/advance\_search

Body :

* min\_order : text
* rating\_min : text
* rating\_max : text
* province: text (id)
* city : text (id)
* courier\_id : text (id)
* price\_min : text
* price\_max: text

Output:

* Status:true, product\_info :[] ,count,count\_all





1. SortByAsc

Route : http://localhost/dress\_marketplace/api/sort\_by\_asc

Body :

Output:

* Status:true, product\_info :[]

1. SortByDesc

Route : http://localhost/dress\_marketplace/api/sort\_by\_desc

Body :

Output:

* Status:true, product\_info :[]

1. Add To Favorite

Route : http://localhost/dress\_marketplace/api/add\_to\_favorite

Body :

* token : text
* store\_id : text

Output:

* Status:true, message : "Favorite Store Successfully"

1. UnFavorite

Route : http://localhost/dress\_marketplace/api/delete\_from\_favorite

Body :

* token : text
* store\_id : text

Output:

- "status": true,

- "message": "Remove Favorite Store Successfully"

1. My Favorite Store

Route : http://localhost/dress\_marketplace/api/my\_favorite

Body :

* token : text

Output:

- "status": true,

- "result": []

1. Get Sort By Id

Route : http://localhost/dress\_marketplace/api/get\_sort\_by\_id

Body :

* sort\_id : text (1 = Recommended, 2=Newest,3=latest)

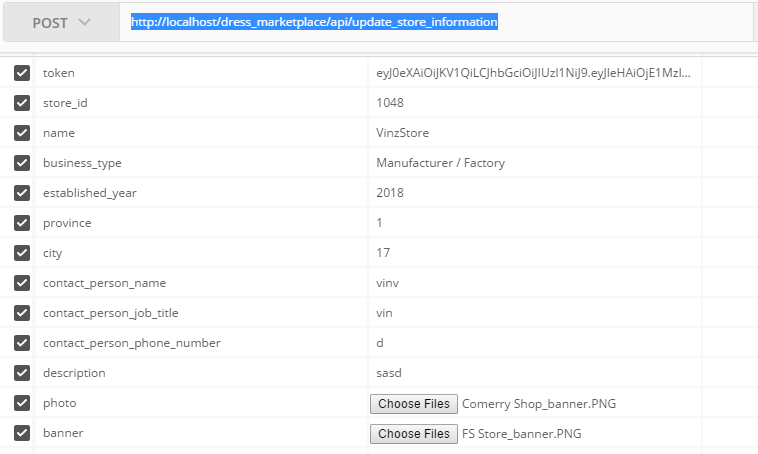
Output:

- "status": true,

- "product\_info": []

1. Update Store Information

Route : http://localhost/dress\_marketplace/api/update\_store\_information



{

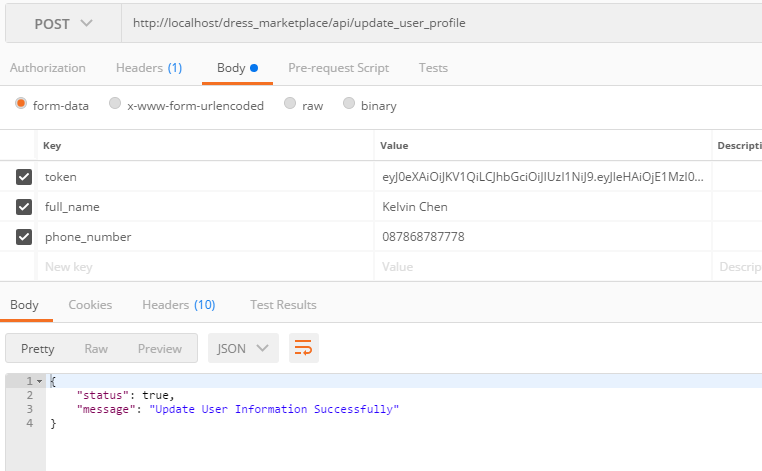
"status": true,

"message": "Update Store Information Successfully"

}

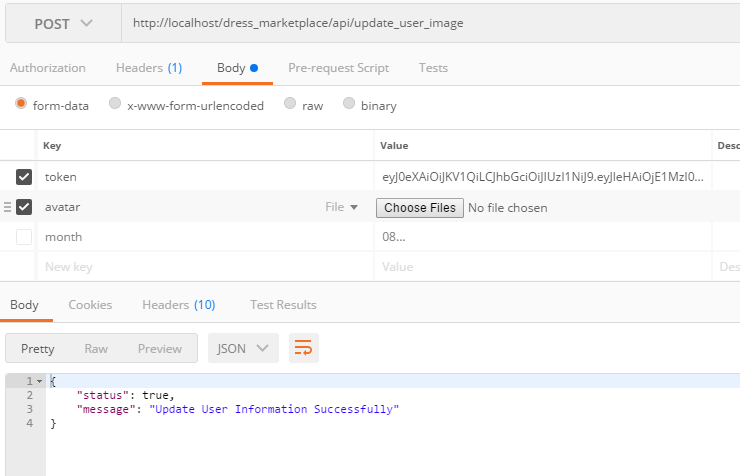
1. Update User Profile

POST <http://localhost/dress_marketplace/api/update_user_profile>



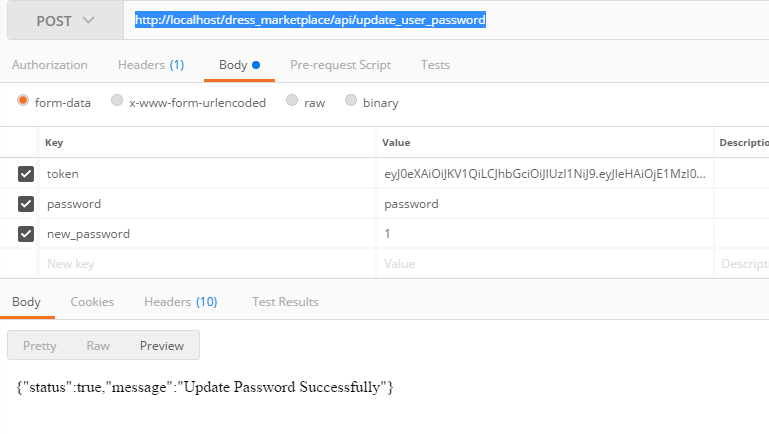
1. Update User Image

POST http://localhost/dress\_marketplace/api/update\_user\_image



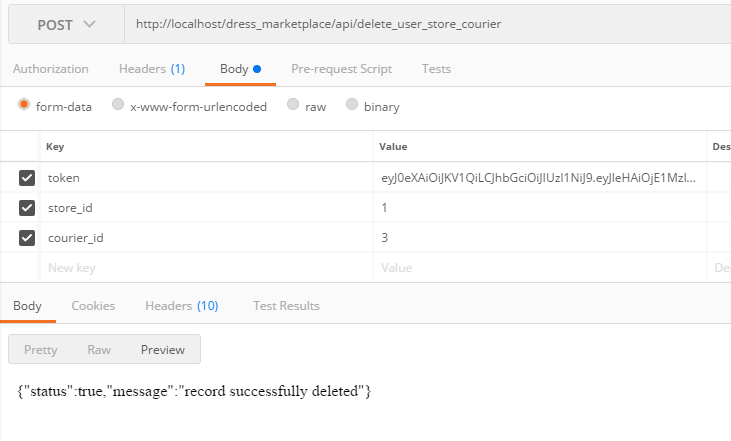
1. Update User Password

POST : <http://localhost/dress_marketplace/api/update_user_password>

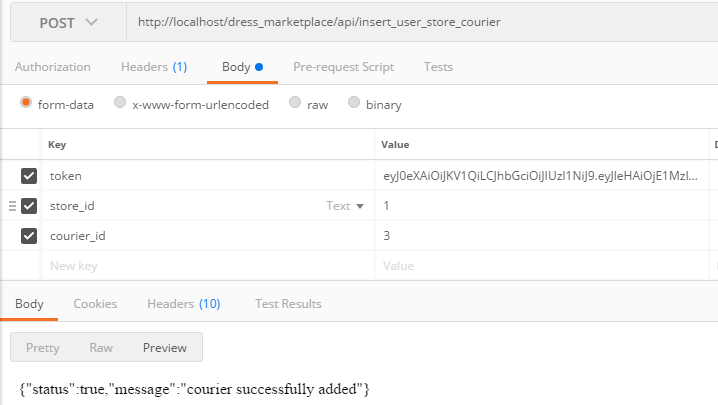


1. Delete Courier Store

POST <http://localhost/dress_marketplace/api/delete_user_store_courier>

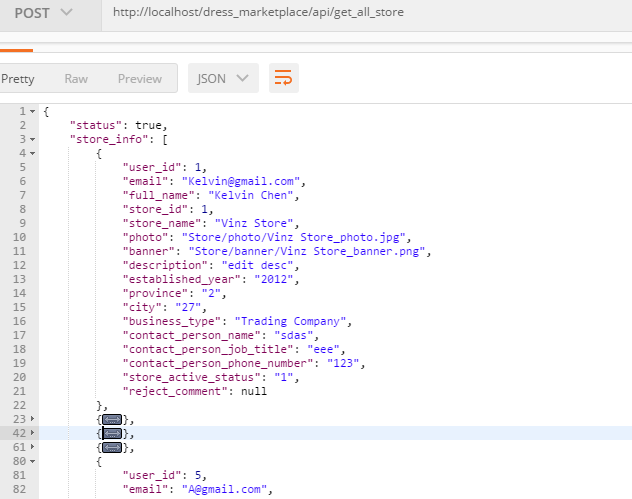


1. Insert Courier Store



1. Get All Store

POST : <http://localhost/dress_marketplace/api/get_all_store>

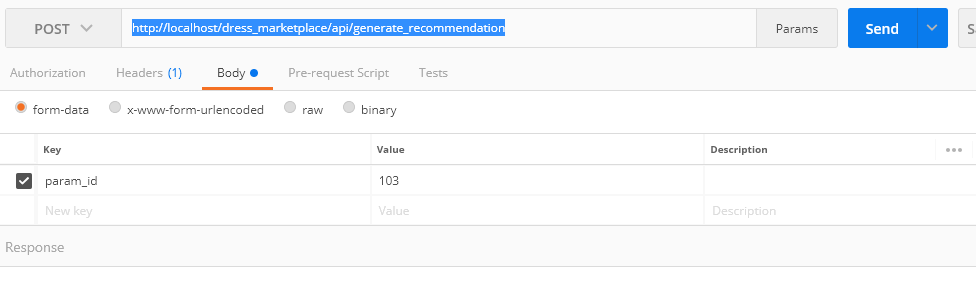


1. Generate Recommendation

POST : <http://localhost/dress_marketplace/api/generate_recommendation>

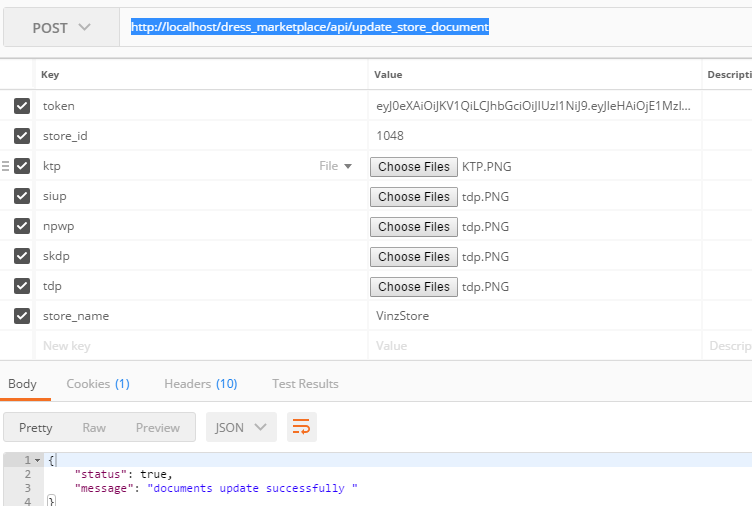
Body :

* Param\_id : text (isi 103)



1. Update Document Supporting

<http://localhost/dress_marketplace/api/update_store_document>

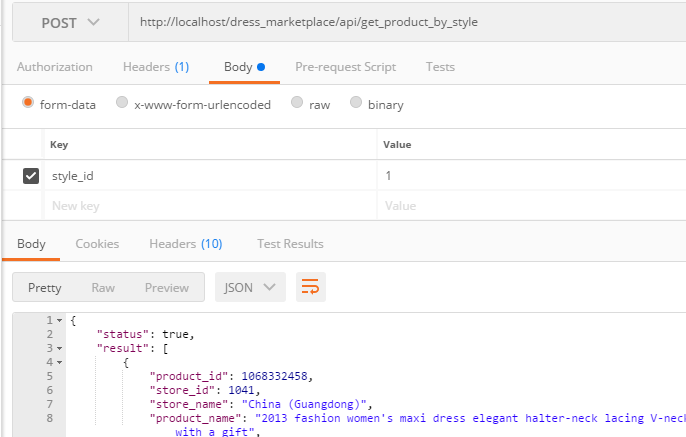


Yg update store information uda w edit diatas

1. Get Product By Style

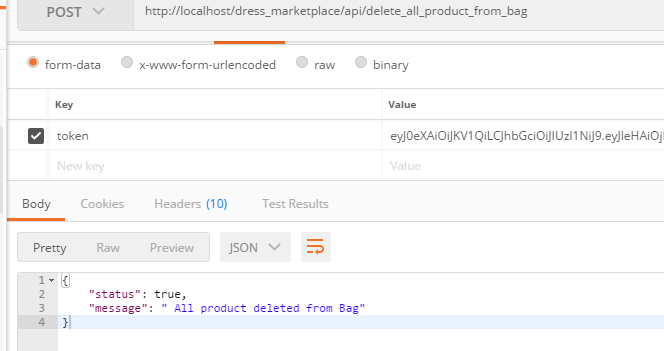
<http://localhost/dress_marketplace/api/get_product_by_style>

Body : style\_id



1. Delete All product From BAg

http://localhost/dress\_marketplace/api/delete\_all\_product\_from\_bag



Keterangan :

Yg advanced search rsultnya w kurangin yg g penting g ditmapilkn

1. Seller Panel Dashboard

Route : http://localhost/dress\_marketplace/api/dashboard

Body :

* token : text
* type : text (pilih salah satu diantara 1,2,3)

keterangan type:

1 = today (berarti total transaksi hari ini)

2 = year to date (berarti total transaksi selama tahun ini)

3 = month to date (berarti total transaksi selama bulan ini)

Output:

* Status:true, result : []

1. Delete Product

Route : http://localhost/dress\_marketplace/api/delete\_product

Body :

* product\_id : text

Output:

* Status:true, message : “Product Deleted”

Berhubung permintaan dari Se Arwin, katanya Store baru boleh benar” didaftarkan jika informasi sudah disubmit semua, jadi semua digabung di Register Store saja, gak perlu Register Store Name lagi.

1. Register Store (Revisi)

Route : http://localhost/dress\_marketplace/api//register\_store

Method : POST

Body :

* token : text
* store\_name : text
* photo : file
* banner : file
* description : text
* established\_year : text
* province : text => berupa id yang didapat dari get province diatas
* city : text => berupa id yang didapat dari get city diatas
* business\_type : text
* contact\_person\_name : text (boleh null)
* contact\_person\_job\_title : text (boleh null)
* contact\_person\_phone\_number : text (boleh null)
* courier : text => berupa array berisi id yang didapat dari get courier list
* ktp : file
* siup : file
* npwp : file
* skdp : file
* tdp : file

output:

* ‘status’ => false, ‘message’ => ‘error : blabla’
* ‘status’ => true, ‘message’ => ‘Store registered successfully’