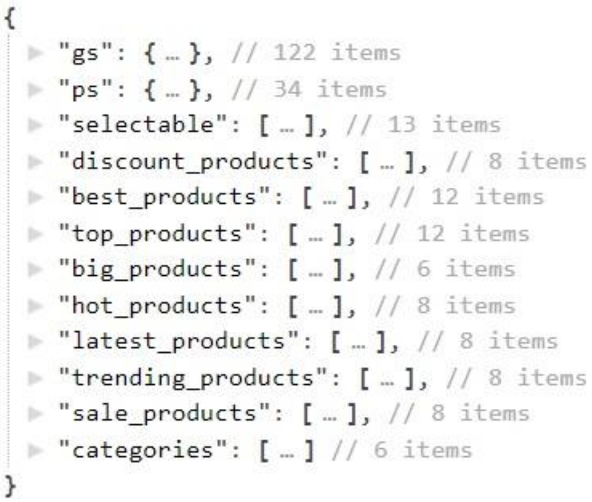
1)

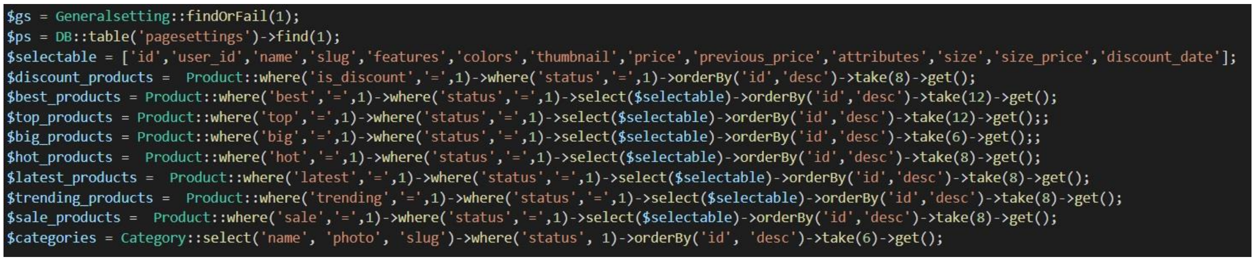
endpoint: /

method: GET

body:



```
{
  ▶ "gs": { … }, // 122 items
  ▶ "ps": { … }, // 34 items
  ▶ "selectable": [ … ], // 13 items
  ▶ "discount_products": [ … ], // 8 items
  ▶ "best_products": [ … ], // 12 items
  ▶ "top_products": [ … ], // 12 items
  ▶ "big_products": [ … ], // 6 items
  ▶ "hot_products": [ … ], // 8 items
  ▶ "latest_products": [ … ], // 8 items
  ▶ "trending_products": [ … ], // 8 items
  ▶ "sale_products": [ … ], // 8 items
  ▶ "categories": [ … ] // 6 items
}
```



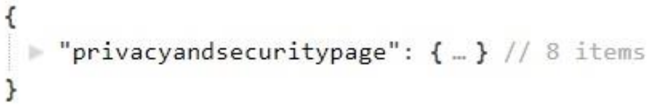
```
$gs = Generalsetting::findOrFail(1);
$ps = DB::table('pagesettings')->find(1);
$selectable = ['id','user_id','name','slug','features','colors','thumbnail','price','previous_price','attributes','size','size_price','discount_date'];
$discount_products =  Product::where('is_discount','=',1)->where('status','=',1)->orderBy('id','desc')->take(8)->get();
$best_products = Product::where('best','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(12)->get();
$top_products = Product::where('top','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(12)->get();;
$big_products = Product::where('big','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(6)->get();;
$hot_products =  Product::where('hot','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(8)->get();
$latest_products =  Product::where('latest','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(8)->get();
$trending_products =  Product::where('trending','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(8)->get();
$sale_products =  Product::where('sale','=',1)->where('status','=',1)->select($selectable)->orderBy('id','desc')->take(8)->get();
$categories = Category::select('name', 'photo', 'slug')->where('status', 1)->orderBy('id', 'desc')->take(6)->get();
```

2)

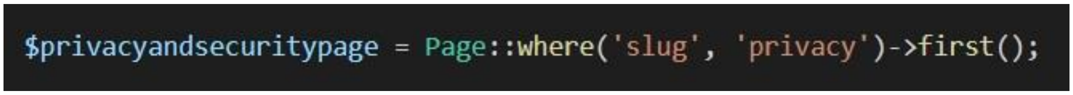
endpoint: /privacyandsecurity

method: GET

body:



```
{
  ▶ "privacyandsecuritypage": { … } // 8 items
}
```



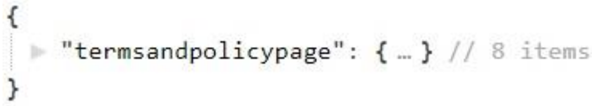
```
$privacyandsecuritypage = Page::where('slug', 'privacy')->first();
```

3)

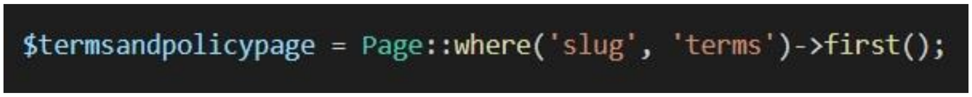
endpoint: /termsandpolicy

method: GET

body:



```
{
  ▶ "termsandpolicypage": { … } // 8 items
}
```



```
$termsandpolicypage = Page::where('slug', 'terms')->first();
```

4)

endpoint: /register

method: POST

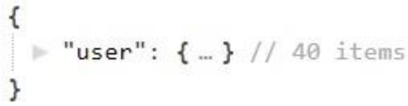
body: { “name”: “testing”, “email”:”test@testing.com”,”password”:”1212121212121” }

5)

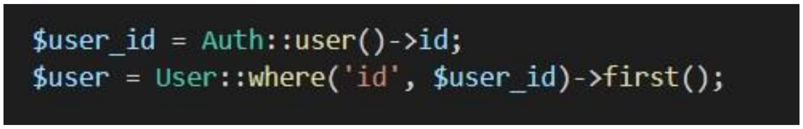
endpoint: /user

method: GET

body:



```
{
  ▶ "user": { … } // 40 items
}
```



```
$user_id = Auth::user()->id;
$user = User::where('id', $user_id)->first();
```

6)

endpoint: /login

method: POST

body: { “email”:”test@testing.com”,”password”:”1212121212121” }

7)

endpoint: /forgot

method: POST

body: { “email”:”test@testing.com” }

8)

endpoint: /logout

method: GET

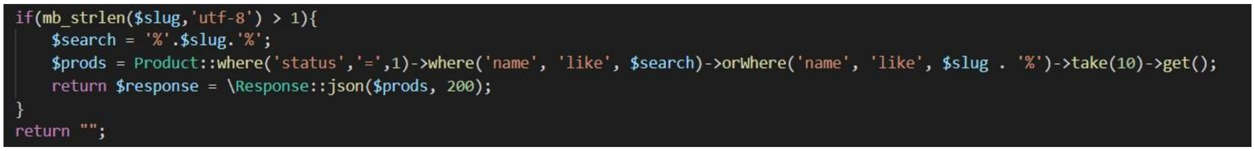
body: { }

9)

endpoint: /autosearch/product/{slug}

method: GET

body: { “prods”: {…} }



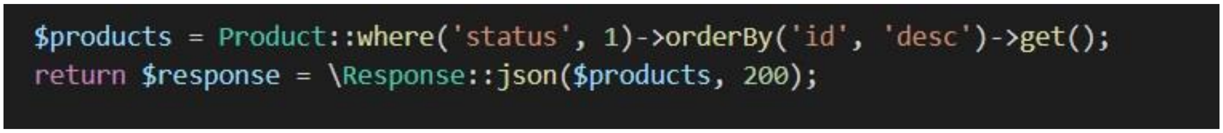
```php
if(mb_strlen($slug,'utf-8') > 1){
    $search = '%'.$slug.'%';
    $prods = Product::where('status','=',1)->where('name', 'like', $search)->orWhere('name', 'like', $slug . '%')->take(10)->get();
    return $response = \Response::json($prods, 200);
}
return "";
```

10)

endpoint: /allproducts

method: GET

body: { “products”: {…} }



```php
$products = Product::where('status', 1)->orderBy('id', 'desc')->get();
return $response = \Response::json($products, 200);
```

11)

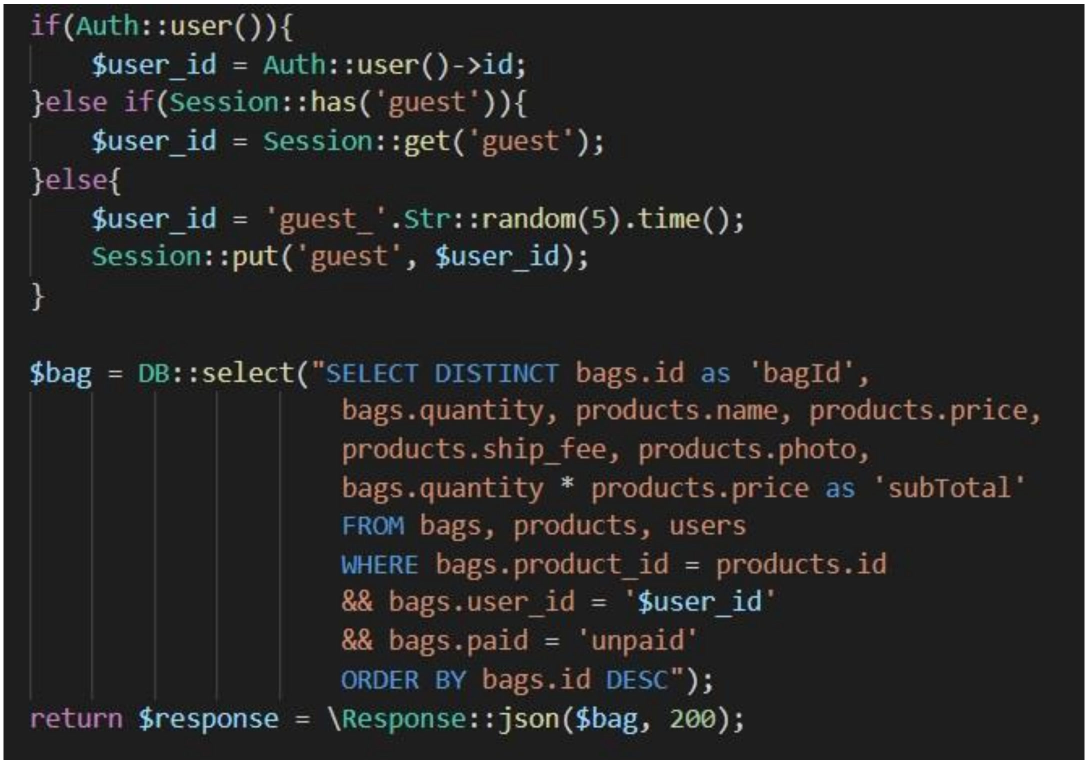
endpoint: /cart

method: GET

body: [

{"bagId": 88,"quantity": "1","name": "MILO SACHET","price": 45, "ship_fee": "0","photo": "16050173864aOKdu9k.jpg","subTotal": 45},{"bagId": 87, "quantity": "1","name": "DANGOTE SUGAR 50kg","price": 18500, "ship_fee": "0","photo": "1606283463IYWu27gf.jpg","subTotal": 18500}

]



```
if(Auth::user()){
    $user_id = Auth::user()->id;
}else if(Session::has('guest')){
    $user_id = Session::get('guest');
}else{
    $user_id = 'guest_'.Str::random(5).time();
    Session::put('guest', $user_id);
}

$bag = DB::select("SELECT DISTINCT bags.id as 'bagId',
                    bags.quantity, products.name, products.price,
                    products.ship_fee, products.photo,
                    bags.quantity * products.price as 'subTotal'
                    FROM bags, products, users
                    WHERE bags.product_id = products.id
                    && bags.user_id = '$user_id'
                    && bags.paid = 'unpaid'
                    ORDER BY bags.id DESC");
return $response = \Response::json($bag, 200);
```

12)

endpoint: /checkout

method: GET

body: { }

13)

endpoint: /addtobag

method: POST

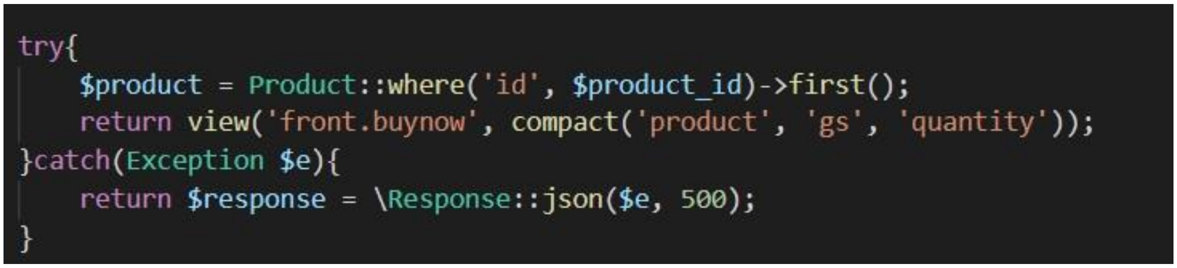
body: { “product_id”:23,”user_id”:11,”user_type”:’’guest”,”quantity”:23,”paid”:”unpaid” }

14)

endpoint: /buynow

method: POST

body: { “product_id”:23, “quantity”:10 }



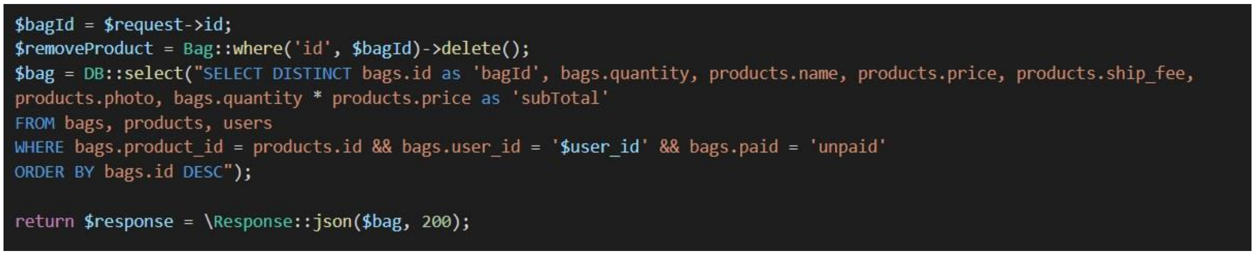
```
try{
    $product = Product::where('id', $product_id)->first();
    return view('front.buynow', compact('product', 'gs', 'quantity'));
}catch(Exception $e){
    return $response = \Response::json($e, 500);
}
```

15)

endpoint: /removeproduct/{id}

method: GET

body: { “id”:23}



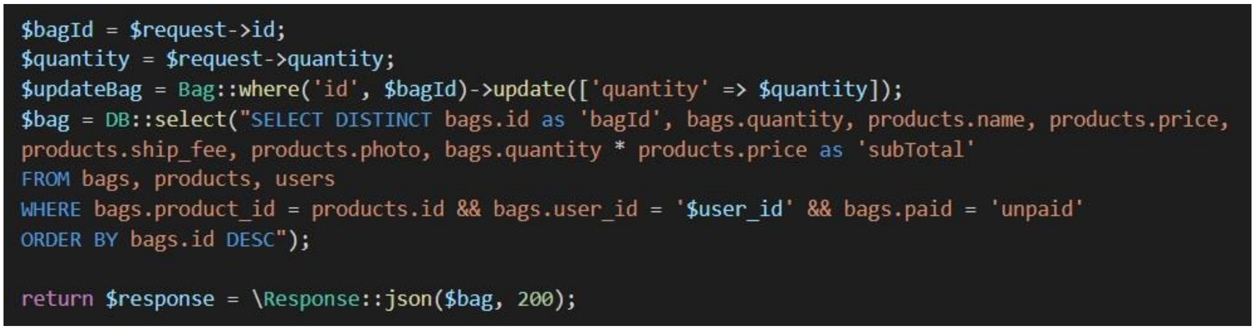
```
$bagId = $request->id;
$removeProduct = Bag::where('id', $bagId)->delete();
$bag = DB::select("SELECT DISTINCT bags.id as 'bagId', bags.quantity, products.name, products.price, products.ship_fee,
products.photo, bags.quantity * products.price as 'subTotal'
FROM bags, products, users
WHERE bags.product_id = products.id && bags.user_id = '$user_id' && bags.paid = 'unpaid'
ORDER BY bags.id DESC");

return $response = \Response::json($bag, 200);
```

16)

endpoint: /addbyone/{id}/{quantity}

method: POST

body: { “id”:23, “quantity”:3}



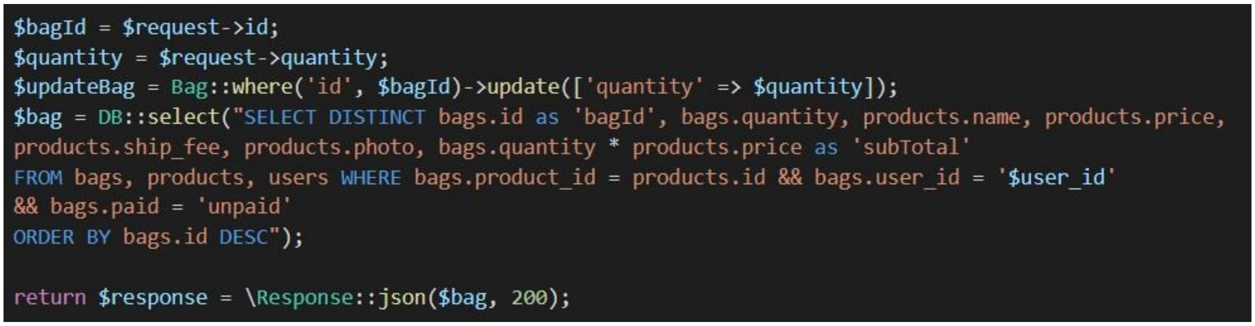
```
$bagId = $request->id;
$quantity = $request->quantity;
$updateBag = Bag::where('id', $bagId)->update(['quantity' => $quantity]);
$bag = DB::select("SELECT DISTINCT bags.id as 'bagId', bags.quantity, products.name, products.price,
products.ship_fee, products.photo, bags.quantity * products.price as 'subTotal'
FROM bags, products, users
WHERE bags.product_id = products.id && bags.user_id = '$user_id' && bags.paid = 'unpaid'
ORDER BY bags.id DESC");

return $response = \Response::json($bag, 200);
```

17)

endpoint: /reducebyone/{id}/{quantity}

method: POST

body: { “id”:23, “quantity”:2}



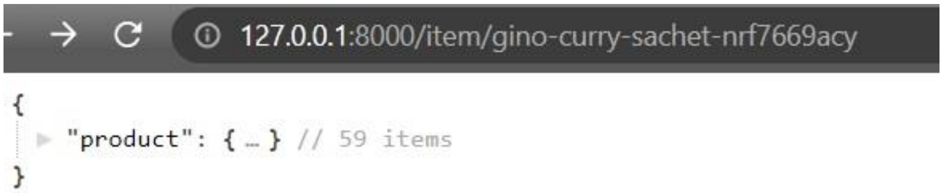
```
$bagId = $request->id;
$quantity = $request->quantity;
$updateBag = Bag::where('id', $bagId)->update(['quantity' => $quantity]);
$bag = DB::select("SELECT DISTINCT bags.id as 'bagId', bags.quantity, products.name, products.price,
products.ship_fee, products.photo, bags.quantity * products.price as 'subTotal'
FROM bags, products, users WHERE bags.product_id = products.id && bags.user_id = '$user_id'
&& bags.paid = 'unpaid'
ORDER BY bags.id DESC");

return $response = \Response::json($bag, 200);
```

18)

endpoint: /item/{slug}

method: GET

body: { “slug”:”diapers-nhv05205kl” }



127.0.0.1:8000/item/gino-curry-sachet-nrf7669acy

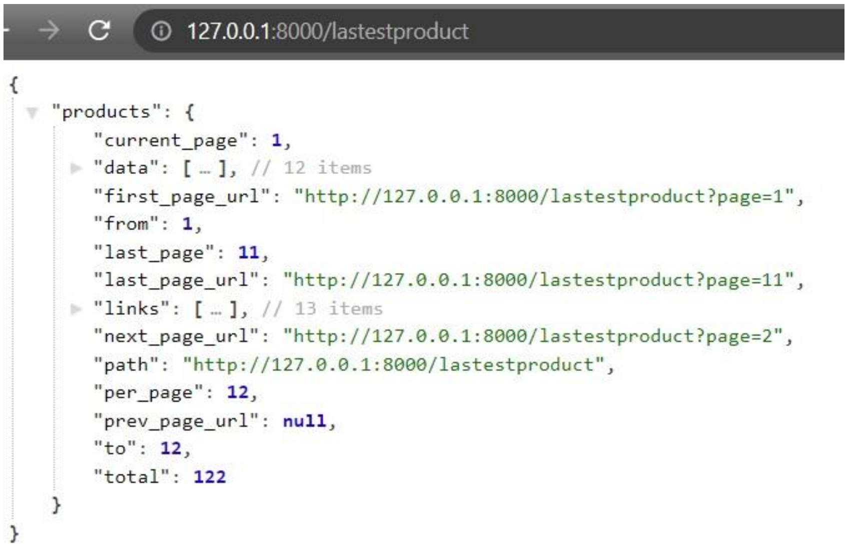
```
{
  ▶ "product": { … } // 59 items
}
```

19)

endpoint: /lastestproduct

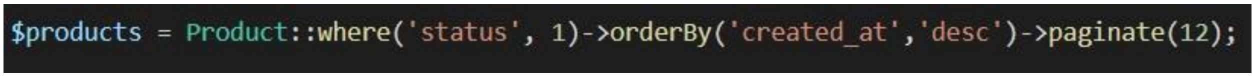
method: GET

body:



127.0.0.1:8000/lastestproduct

```
{
  ▼ "products": {
        "current_page": 1,
      ▶ "data": [ … ], // 12 items
        "first_page_url": "http://127.0.0.1:8000/lastestproduct?page=1",
        "from": 1,
        "last_page": 11,
        "last_page_url": "http://127.0.0.1:8000/lastestproduct?page=11",
      ▶ "links": [ … ], // 13 items
        "next_page_url": "http://127.0.0.1:8000/lastestproduct?page=2",
        "path": "http://127.0.0.1:8000/lastestproduct",
        "per_page": 12,
        "prev_page_url": null,
        "to": 12,
        "total": 122
    }
}
```



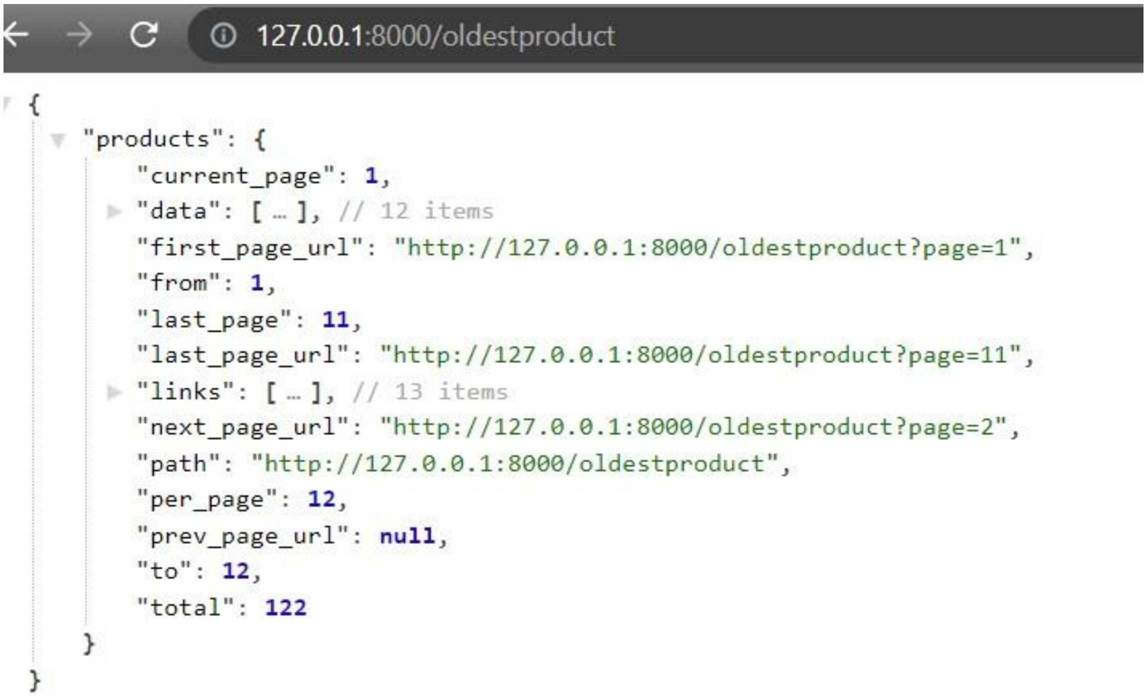
```
$products = Product::where('status', 1)->orderBy('created_at','desc')->paginate(12);
```

20)

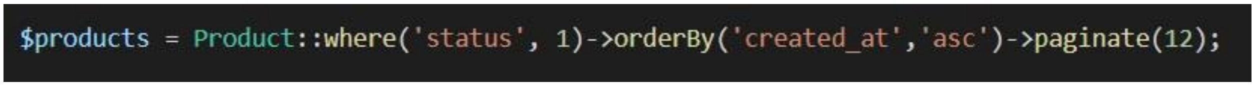
endpoint: /oldestproduct

method: GET

body:



```
{
  "products": {
      "current_page": 1,
    ▶ "data": [ … ], // 12 items
      "first_page_url": "http://127.0.0.1:8000/oldestproduct?page=1",
      "from": 1,
      "last_page": 11,
      "last_page_url": "http://127.0.0.1:8000/oldestproduct?page=11",
    ▶ "links": [ … ], // 13 items
      "next_page_url": "http://127.0.0.1:8000/oldestproduct?page=2",
      "path": "http://127.0.0.1:8000/oldestproduct",
      "per_page": 12,
      "prev_page_url": null,
      "to": 12,
      "total": 122
  }
}
```



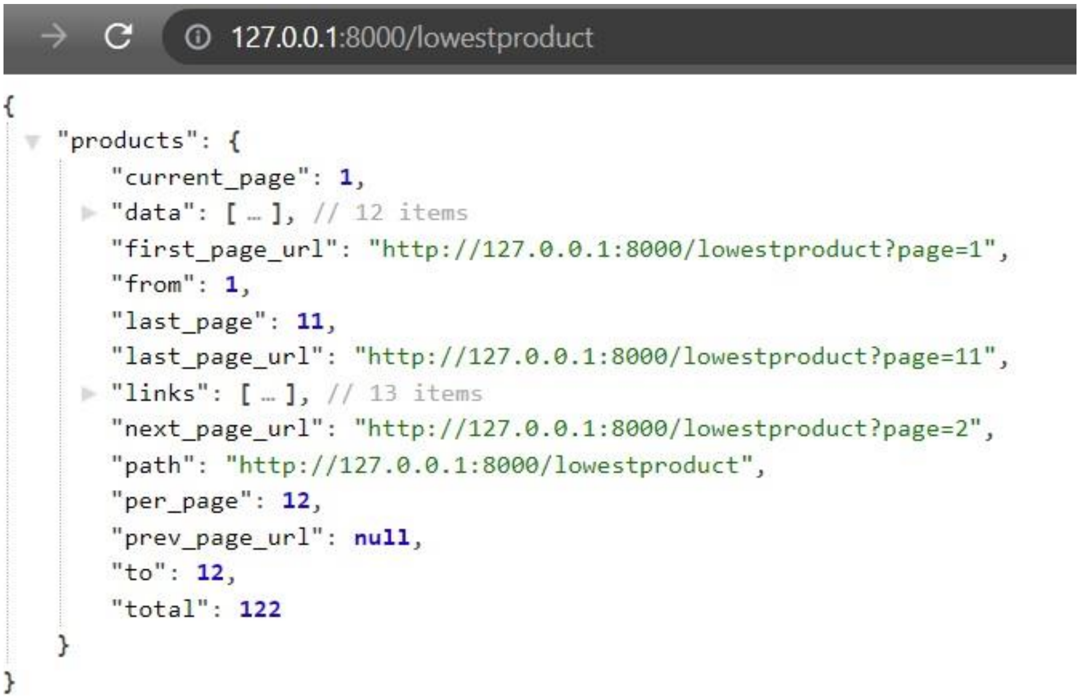
```
$products = Product::where('status', 1)->orderBy('created_at','asc')->paginate(12);
```

21)

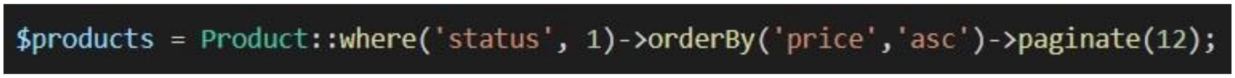
endpoint: /lowestproduct

method: GET

body:



```
{
  "products": {
      "current_page": 1,
      "data": [ … ], // 12 items
      "first_page_url": "http://127.0.0.1:8000/lowestproduct?page=1",
      "from": 1,
      "last_page": 11,
      "last_page_url": "http://127.0.0.1:8000/lowestproduct?page=11",
      "links": [ … ], // 13 items
      "next_page_url": "http://127.0.0.1:8000/lowestproduct?page=2",
      "path": "http://127.0.0.1:8000/lowestproduct",
      "per_page": 12,
      "prev_page_url": null,
      "to": 12,
      "total": 122
  }
}
```



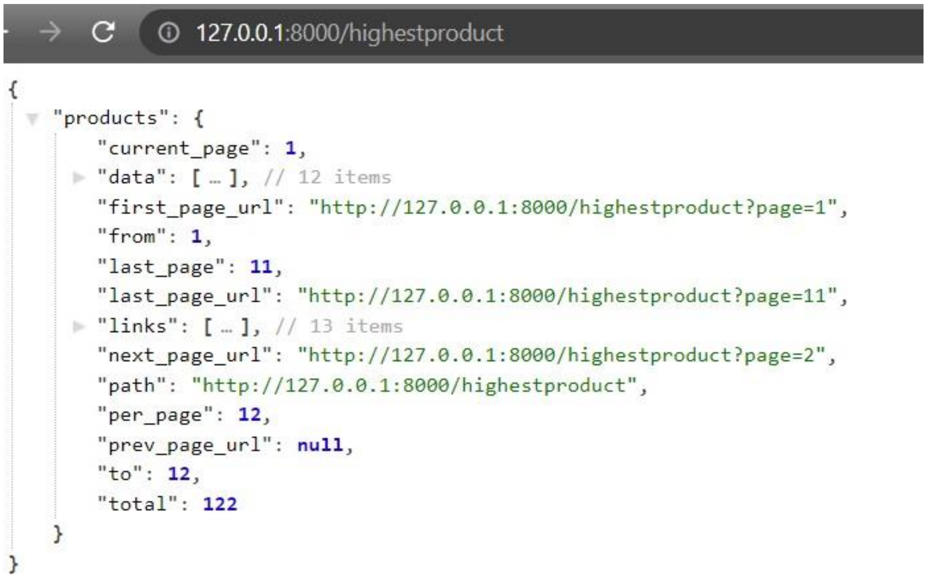
```
$products = Product::where('status', 1)->orderBy('price','asc')->paginate(12);
```

22)

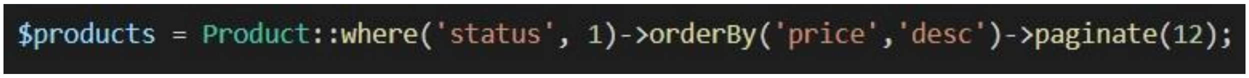
endpoint: /highestproduct

method: GET

body:



```
{
  "products": {
      "current_page": 1,
      "data": [ … ], // 12 items
      "first_page_url": "http://127.0.0.1:8000/highestproduct?page=1",
      "from": 1,
      "last_page": 11,
      "last_page_url": "http://127.0.0.1:8000/highestproduct?page=11",
      "links": [ … ], // 13 items
      "next_page_url": "http://127.0.0.1:8000/highestproduct?page=2",
      "path": "http://127.0.0.1:8000/highestproduct",
      "per_page": 12,
      "prev_page_url": null,
      "to": 12,
      "total": 122
  }
}
```



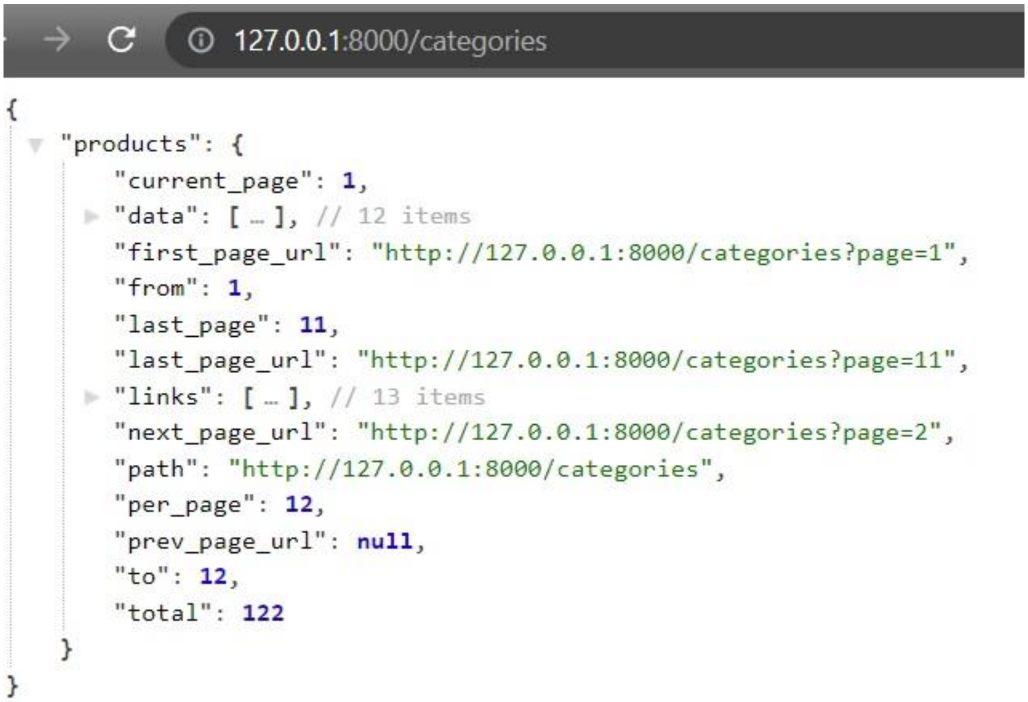
```
$products = Product::where('status', 1)->orderBy('price','desc')->paginate(12);
```

23)

endpoint: /categories

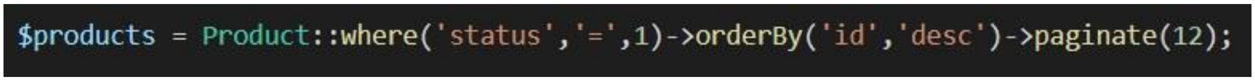
method: GET

body:



127.0.0.1:8000/categories

```
{
  ▼ "products": {
        "current_page": 1,
      ▶ "data": [ … ], // 12 items
        "first_page_url": "http://127.0.0.1:8000/categories?page=1",
        "from": 1,
        "last_page": 11,
        "last_page_url": "http://127.0.0.1:8000/categories?page=11",
      ▶ "links": [ … ], // 13 items
        "next_page_url": "http://127.0.0.1:8000/categories?page=2",
        "path": "http://127.0.0.1:8000/categories",
        "per_page": 12,
        "prev_page_url": null,
        "to": 12,
        "total": 122
    }
}
```



```
$products = Product::where('status','=',1)->orderBy('id','desc')->paginate(12);
```

24)

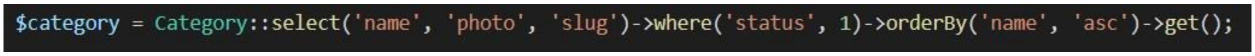
endpoint: /allcategories

method: GET

body: [

{"name": "Cleaning & Toiletries","photo": "16378632131636785614household.png","slug": "cleaning--toiletries"},{"name": "Drinks & Beverages","photo":"163786318516367858671636785347groceries.png","slug": "Beverages"},{"name": "Electronic","photo": "16378633041636786232electronics.png","slug": "electric"},{"name": "Smart Phones & Tablet","photo": "16378632831636786425smart-phones.png","slug": "smart-phones-and-tablets"}

]



```
$category = Category::select('name', 'photo', 'slug')->where('status', 1)->orderBy('name', 'asc')->get();
```

25)

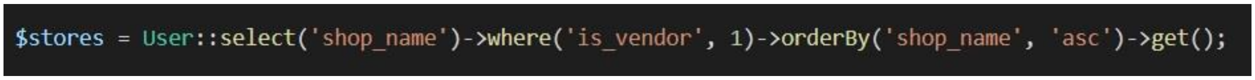
endpoint: /allstores

method: GET

body: [

{"shop_name": "DABAGI"},{"shop_name": "Kahioja Stores"}

]



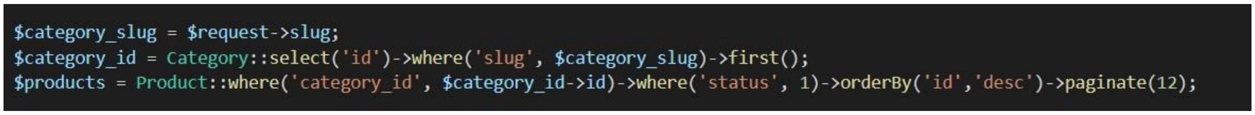
```
$stores = User::select('shop_name')->where('is_vendor', 1)->orderBy('shop_name', 'asc')->get();
```

26)

endpoint: /category/{slug}

method: GET

body: { “slug”: “electric” }



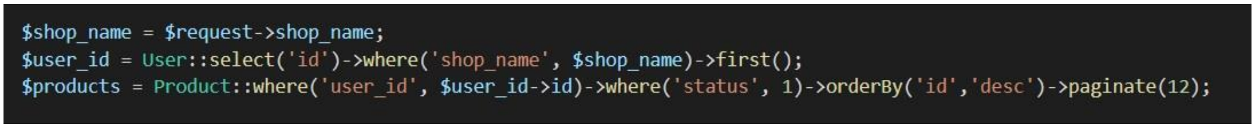
```
$category_slug = $request->slug;
$category_id = Category::select('id')->where('slug', $category_slug)->first();
$products = Product::where('category_id', $category_id->id)->where('status', 1)->orderBy('id','desc')->paginate(12);
```

27)

endpoint: /store/{shop_name}

method: GET

body: { “shop”: “kahioja” }



```
$shop_name = $request->shop_name;
$user_id = User::select('id')->where('shop_name', $shop_name)->first();
$products = Product::where('user_id', $user_id->id)->where('status', 1)->orderBy('id','desc')->paginate(12);
```

28)

endpoint: /flutterwave/submit

method: POST

body: { “amount”: 5000, “email”:”testing@test.com”,”phone_number”:07037645413,”name”:”piccolo”
}

29)

endpoint: /flutterwave/callback

method: GET

body: { … }

30)

endpoint: /checkoutfailed

method: GET

body: { … }

31)

endpoint: /checkoutsuccess

method: GET

body: { … }

We will be glad if you can check the method in each controller to see how the logic was created. Thanks a lot.

Best regards,

Team Piccolo