Sale-a-Bration Endpoints

Host: <http://saleabration.herokuapp.com>

**Brands**

PATH /brands

Returns a list of all brands

Schema: <https://api.stoplight.io/v1/versions/qiw2mX8rkZXqkX4dd/export/oas.json>

Example: <http://saleabration.herokuapp.com/brands>

Response:

[

{

"brand": "adf",

"slug": "adf",

"title": "ADF"

},

{

"brand": "aveda",

"slug": "aveda",

"title": "Aveda"

}, { … }

]

**Products**

PATH /brand

Returns product lists for each brand

Schema: <https://api.stoplight.io/v1/versions/Nmbt5qouBoirBYciD/export/oas.json>

Endpoints:

GET /adf

GET /bobbibrown

GET /bumbleandbumble

GET /clinique

GET /darphin

GET /esteelauder

GET /jomalone

GET /m-a-c

GET /origins

GET /rodin

GET /smashbox

GET /tomford

Example: <http://saleabration.herokuapp.com/brand/m-a-c>

Response:

[

{

"shadeOrSize": "Quiet Time",

"formattedPrice": "$13",

"formattedSale": "$0.00",

"sale": 0,

"brand": "m-a-c",

"id": "m-a-c0",

"qty": 1,

"parent": {

"prodRgnName": "M·A·C Studio Nail Lacquer",

"largeImage": "http://www.maccosmetics.com/media/export/cms/products/640x600/mac\_sku\_MW8919\_640x600\_0.jpg"

}

}, { … }

]

**Orders**

PATH /orders

Endpoint for confirmed orders. Works with the “orders” Postgres table.

Schema: <https://api.stoplight.io/v1/versions/86Nit5ox2J8dbB8R4/export/oas.json>

POST /confirm/:userid

Save user’s confirmed shopping list by copying user’s entry from “lists” table over to “orders” table. Alternatively, receive an array of items in the body of this request.

GET /get/:userid

Get user’s confirmed shopping list

GET /total/:userid

Return the total item quantity and subtotal

GET /export

Returns an array of CSV files on the server, each containing a user’s cart information from the “orders” table. The filename is the first part of their login email (e.g. “jlalor” for jlalor@estee.com). There is also an “all.csv” file, containing all users’ entries. You can access the files at: **saleabration.herokuapp.com/orders/filename.csv**

GET /export/:emailname

Export individual user’s cart / wishlist to a CSV. :emailname param is the same as described above (e.g. “jlalor” for jlalor@estee.com). (This endpoint is still in development.)

**Lists**

PATH /lists

Endpoint for updating shopping lists and wishlists. Works with the “lists” Postgres table. NOTE: This group of endpoints is not used – except **/get/:userid**, which is used for data type configuration in Dropsource – however, they may prove useful in the future.

Schema: <https://api.stoplight.io/v1/versions/fz2RoJwDpvNFznkG4/export/oas.json>

**POST /new/:userid**

Creates a new entry in the “lists” table

The :userid parameter is equivalent to the Firebase uid. This is added to the “lists” table once a user logs in for the first time. See Dropsource documentation for the Sign in with Firebase Authentication action. This action returns Firebase uid from the specified Firebase project. The body’s POST data should contain an object specifying the firstname and email.

Example: [http://saleabration.herokuapp.com/lists/new/cfyANJ3EhUa8fZCeSRN4059Z8rg2](http://saleabration.herokuapp.com/lists/add/cfyANJ3EhUa8fZCeSRN4059Z8rg2)

BODY:

{ "firstname": "jason", "email": "jasonlalor@gmail.com" }

Response (200; success):

{

"userID": " cfyANJ3EhUa8fZCeSRN4059Z8rg2",

"firstname": "jason",

"email": "jasonlalor@gmail.com"

}

Response (404; if user already exists):

{

"error": "User already exists"

}

**GET /get/:userid**

Return saved cart / wishlist for this user

Example: [http://saleabration.herokuapp.com/lists/get/cfyANJ3EhUa8fZCeSRN4059Z8rg2](http://saleabration.herokuapp.com/lists/add/cfyANJ3EhUa8fZCeSRN4059Z8rg2)

Response:

[

{

"id": "5H1A01",

"sale": 0,

"qty": 1,

"brand": "featured",

"formattedPrice": "$60.00",

"formattedSale": "$0.00",

"shadeOrSize": " ",

"parent": {

"prodRgnName": "Aramis Classic EDT",

"largeImage": "http://ota.esteeonline.com/wishlist/images/featured/aramis-classic-edt.jpeg"

}

}, { … }

]

**POST /add/:userid**

Add or remove item from the user’s cart / wishlist. The body’s POST data should contain the item’s object data. If the item is already in the cart, it is removed. Otherwise, it is added.

Example: <http://saleabration.herokuapp.com/lists/add/cfyANJ3EhUa8fZCeSRN4059Z8rg2>

BODY:

{

"id":"2M8E01",

"sale":0,

"brand":"adf",

"qty":1,

"formattedPrice":"$67",

"formattedSale":"$0.00",

"shadeOrSize":" ",

"parent":{

"prodRgnName":"Aramis Havana Cologne",

"largeImage":"http://ota.esteeonline.com/wishlist/images/adf/aramis-havana-cologne.jpeg"

}

}

Response:

[

{

"user": "cfyANJ3EhUa8fZCeSRN4059Z8rg2",

"item": "2M8E01",

"status": "Added"

}

]

OR

[

{

"user": "cfyANJ3EhUa8fZCeSRN4059Z8rg2",

"item": "2M8E01",

"status": "Deleted"

}

]

**POST /add/:userid/:qty**

Update the quantity for a specific item in the user’s list. The body’s POST data should contain the item’s object data. The :qty parameter should contain the new quantity for the specified item.

**GET /count/:userid**

Returns the total quantity and subtotal for the specified user’s list.

**GET /lists/sort/:userid/:sorted/:reverse**

Returns the list for the specified userid, sorted by a key value (:sorted, e.g. “sale” or “brand”) and in ascending/descending order (:reverse, with a value of either “asc” or “desc”).

Example: <http://saleabration.herokuapp.com/lists/sort/cfyANJ3EhUa8fZCeSRN4059Z8rg2/brand/asc>