CommentSold Mobile Code Test

This document outlines a code evaluation which will test your ability to put together a simple mobile application.

There are a few core components that you will be expected to implement and some room for your own approach to tackling the problem.

Requirements

The goal of our project is to build a small mobile app that can provide users with the ability to view and edit their inventory system.

This system is made up of the following resources:

Users Used for authentication and to scope the available data, all other records are tied to products which the authenticated user account owns.

Products Products represent items in the system that can be sold, they are linked to both inventory which are specific variants of the products and orders which are individual orders for the product and inventory item.

Inventory Inventory represent a variation of a product with an associated color, size, quantity, sku along with the cost and size of the item.

Orders Orders are instances of purchases by an individual user, they tie together product and inventory records and provide the customer information along with the status of the order in the system (eg. whether it has been shipped).

Features

Our mobile app has a few key features that we can provide to the end user.

The goal of this exercise is to provide a subset of the features outlined below using the API endpoints provided you do not have to implement all features outlined here.

Your mobile app should implement the following:

1. Authentication we should provide the user with a way to authenticate this allows them to enter their email and password and then loads the data associated with their account.
    - Under the hood you should make a call to the status endpoint with the user email and password the response you get back will include a JWT token which can be used for all subsequent calls.
2. Product Interface: The app should provide a user interface to list, view the details of, create, edit or delete product records and should allow them to navigate through their whole set of products.

Optional Features:

1. Inventory Interface: The app can provide the user with an interface which lets the user list their inventory records, filter the inventory set based on the color, size or quantity of the item, and optionally to create, edit or delete inventory records.
2. Orders Interface: The app can provide the user with an interface to list the orders related to products/inventory under their account. Orders can be filtered by their order status. An additional optional feature could be to provide a local total of the cost, revenue or totals for the orders being displayed.

API Documentation

Our mobile app should use the following API endpoints to provide information for the users in the system.

NOTE: For all POST, PUT calls the API expects a content type of `application/json` to be set.

NOTE: Pagination starts from 0 so no page specified will return the first page which is page 0, then any subsequent numbers will return the next paged set of data eg, 0, 1, 2, 3, etc

Authentication

The status endpoint will return a valid JWT token for use with all other endpoints.

GET `https://cscodetest.herokuapp.com/api/status`

Request:

The status endpoint expects a basic auth request containing a valid email and password for a user in the system.

(*Provided at the end of this document is a list of valid email/password combinations*)

Sample request using curl:

```
curl -u lekisha.dinatale@foo.com:MFdqzRnhSb https://cscodetest.herokuapp.com/api/status
```

Response:

The status endpoint will return a JWT token value and an error code eg.

```
{
  "error": 0,
  "token": "jwt-token-here"
}
```

Products

Product Object:

```
        "id": 10766,
        "product_name": "Casual Chic Tie",
        "description": "Casual Chic Tie is an amazing Tie Product...Lorem ipsum dolor sit amet, ne est diceret platonem, ius viris fab
ulas ea. Amet elit putent ea sit, per accumsan verterem cu.",
        "style": "Casual Chic",
        "brand": "Unravel",
        "created_at": "2018-05-27T11:03:15Z",
        "updated_at": "2019-01-13T06:49:34Z",
        "url": "",
        "product_type": "clothing",
        "shipping_price": 748,
        "note": "",
```

Products List

GET `https://cscodetest.herokuapp.com/api/products`

```
Parameters:
    page - The result page to return
    limit - The page limit to use, by default this value is 50 results
```

Sample request using curl:

```
curl -H 'Accept: application/json' -H "Authorization: Bearer jwt-token-here" https://cscodetest.herokuapp.com/api/products?limit=2
```

Response:

```
200 Response code
{
"count": 2,
"products": [{
        "id": 10766,
        "product_name": "Casual Chic Tie",
        "description": "Casual Chic Tie is an amazing Tie Product...Lorem ipsum dolor sit amet, ne est diceret platonem, ius viris fab
ulas ea. Amet elit putent ea sit, per accumsan verterem cu.",
        "style": "Casual Chic",
        "brand": "Unravel",
        "created_at": "2018-05-27T11:03:15Z",
        "updated_at": "2019-01-13T06:49:34Z",
        "url": "",
        "product_type": "clothing",
        "shipping_price": 748,
        "note": "",
        "admin_id": 24
    }, {
        "id": 10757,
        "product_name": "Military Cap",
        "description": "Military Cap is an amazing Cap Product...Lorem ipsum dolor sit amet, ne est diceret platonem, ius viris fabula
s ea. Amet elit putent ea sit, per accumsan verterem cu.",
        "style": "Military",
        "brand": "Yummie Tummie",
        "created_at": "2018-04-28T02:45:22Z",
        "updated_at": "2018-08-24T23:21:12Z",
        "url": "",
        "product_type": "clothing",
        "shipping_price": 1313,
        "note": "",
        "admin_id": 24
    }],
    "total": 193
}
```

Product Styles

GET `https://cscodetest.herokuapp.com/api/styles`

Return the set of valid styles for use when creating products.

Sample request:

`curl -H 'Accept: application/json' -H "Authorization: Bearer jwt-token-here" https://cscodetest.herokuapp.com/api/styles`

Response:

```
200 Response code
{
    "styles": ["Streetwear", "Formal Office Wear", "Business Casual", "Evening Black Tie", "Sports Wear",
     "Girly ", "Rocker Chic", "Skateboarders", "Goth", "Maternity", "Hip Hop", "Kawai", "Preppy", "Cowgirl",
      "Lagenlook", "Scene", "Girl next door", "Casual Chic", "Geeky chic", "Military", "Retro", "Flapper",
      "Tomboy", "Garconne look", "Vacation", "Artsy", "Grunge", "Punk", "Boho/Bohemian chic", "Psychedelic",
      "Cosplay", "Haute Couture", "Modest", "Rave", "Flamboyant", "Ankara"]
}
```

The response contains an array of the possible values nested under the `styles` object.

Product Colors

GET `https://cscodetest.herokuapp.com/api/colors`

Return the set of valid colors for use when creating products.

Sample Request:

```
curl -H 'Accept: application/json' -H "Authorization: Bearer jwt-token-here" https://cscodetest.herokuapp.com/api/colors
```

Response:

```
200 Response code
{"colors":["Beige","Black","Blue","Checkered","Cherry Red","Dark Blue","Green","Grey","Light Grey",
"Metallic Red","Orange","Pink","Purple","Steel Blue"]}
```

The response contains an array of the possible values nested under the `colors` object.

Product Item

Get an individual product item's data

GET `https://cscodetest.herokuapp.com/api/product/:id`

Response:

```
Response Code 200
{
  "product": {
    "id": 10852,
    "product_name": "Hip Hop Ear muffs",
    "description": "Hip Hop Ear muffs is an amazing Ear muffs Product...Lorem ipsum dolor sit amet, ne est diceret platonem, ius viris
 fabulas ea. Amet elit putent ea sit, per accumsan verterem cu.",
    "style": "Hip Hop",
    "brand": "The Mountain",
    "created_at": "2018-03-23T08:21:38Z",
    "updated_at": "2018-09-17T07:33:26Z",
    "url": "",
    "product_type": "clothing",
    "shipping_price": 812,
    "note": "",
    "admin_id": 23
  }
}
```

Product Create

POST `https://cscodetest.herokuapp.com/api/product`

Create a new product in the system

```
{
    "name": "Paul's Test Product",
    "description": "This is a test product to try out the API.",
    "style": "Formal Office Wear",
    "brand": "J Crew",
    "shipping_price_cents": 700
}
```

The json payload should include the name, description, style and brand for the product along with some value for the shipping price.

Response:

```
200 Response code
{
    "message": "Created successfully",
    "product_id": 10877
}
```

The response includes a message and the id of the created record.

Product Update

PUT `https://cscodetest.herokuapp.com/api/product/:id`

Update a product in the system. An update can include any of the parameters outlined in the create step the url should include the id of the record in place of `:id`.

Reponse:

```
200 Response code
{
    "message": "Updated successfully",
    "product_id": 10877
}
```

A successful update includes a success message along with the product_id that was updated.

Product Delete

DELETE `https://cscodetest.herokuapp.com/api/product/:id`

Delete a product from the system. Deleting a product can be done by making an HTTP `DELETE` call to the path including the product id.

Response:

A successful deletion will provide a success message and the product_id of the record deleted.

```
200 Response code
{
    "message": "Deleted successfully",
   "product_id": 10876
}
```

In the event of an error you will get 500 error code along with an error message:

```
500 Response code
{
  "error": "Unable to delete product for id: 10877 and user: larhonda.hovis@foo.com"
}
```

Inventory

Inventory Object:

```
  "id": 23897,
  "product_id": 9723,
  "quantity": 16,
  "color": "Black",
  "size": "S",
  "weight": "5",
  "price_cents": 3282,
  "sale_price_cents": 3082,
  "cost_cents": 2982,
  "sku": "OBJPMB",
  "length": "9",
  "width": "2",
  "height": "6",
  "note": ""
```


Inventory List

GET `https://cscodetest.herokuapp.com/api/inventory`

```
Parameters:
    page - The result page to return
    limit - The page limit to use, by default this value is 50 results
    color - The color of the variants to return
    size - The size of the variants to return (XS, S, M, L, XL)
    quantity - The quantity of the variants to return (must specify < or > quantity)
```

eg. `https://cscodetest.herokuapp.com/api/inventory?page=0&limit=50&color=Blue&size=L&quantity=>10`

Response:

```
Response Code 200
{
    "count": 2,
    "inventory": [{
            "id": 23897,
            "product_id": 9723,
            "quantity": 16,
            "color": "Black",
            "size": "S",
            "weight": "5",
            "price_cents": 3282,
            "sale_price_cents": 3082,
            "cost_cents": 2982,
            "sku": "OBJPMB",
            "length": "9",
            "width": "2",
            "height": "6",
            "note": ""
        },
        {
            "id": 23866,
            "product_id": 298,
            "quantity": 31,
            "color": "Orange",
            "size": "XL",
            "weight": "8",
            "price_cents": 1829,
            "sale_price_cents": 1629,
            "cost_cents": 1442,
            "sku": "NEWOLO",
            "length": "4",
            "width": "8",
            "height": "9",
            "note": ""
        }
    ],
    "total": 489
}
```

Inventory Item

Return an individual inventory record.

GET `https://cscodetest.herokuapp.com/api/inventory/:id`

Response:

```
Response Code 200
{
  "inventory": {
    "id": 23897,
    "product_id": 9723,
    "quantity": 16,
    "color": "Black",
    "size": "S",
    "weight": "5",
    "price_cents": 3282,
    "sale_price_cents": 3082,
    "cost_cents": 2982,
    "sku": "OBJPMB",
    "length": "9",
    "width": "2",
    "height": "6",
    "note": ""
  }
}
```

Inventory Item Create

Create a new inventory record. Expects a json POST request with a payload of the form shown below.

POST `https://cscodetest.herokuapp.com/api/inventory`

Payload parameters:

```
{
    "product_id": 7969,
    "quantity": 2,
    "color": "Blue",
    "size": "L",
    "weight": 3.0,
    "price_cents": 1500,
    "sale_price_cents": 1100,
    "cost_cents": 700,
    "sku": "abc123",
    "length": 3.00,
    "width": 4.00,
    "height": 7.00,
    "note": ""
}
```

Response:

```
Response Code 200
{
    "inventory_id": 24001,
    "message": "Created successfully"
}
```

Inventory Item Update

Update an inventory record. The update payload can be any subset of the parameters use for creation.

PUT `https://cscodetest.herokuapp.com/api/inventory/:id`

example payload:

```
{
    "color": "Orange",
    "quantity": 8
}
```

Response:

```
Response Code 200
{
  "inventory_id": 24001,
  "message": "Updated successfully"
}
```

Inventory Item Delete

Delete an item from the inventory. Provide the id as part of the url (replace :id).

DELETE `https://cscodetest.herokuapp.com/api/inventory/:id`

Response:

A successful deletion includes the id of the record deleted and a success message.

```
Response Code 200
{
  "inventory_id": 24001,
  "message": "Deleted successfully"
}
```

Orders API

Orders List

The order list endpoint will return a paginated set of orders tied to the user account. This set can be filtered based on the status of the order, to see available status options you can use the Order Status Options endpoint.

GET `https://cscodetest.herokuapp.com/api/orders?page=0&status=shipped`

```
Parameters:
    page - The result page to return
    limit - The page limit to use, by default this value is 50 results
    status - The order status to limit the result set by (currently only 1 option can be set)
```

Response:

```
{
  "count": 2,
  "order": [
    {
      "id": 110281,
      "product_id": 9723,
      "inventory_id": 23897,
      "street_address": "720 South River Landing Road",
      "apartment": "",
      "city": "Edgewater",
      "state": "MD",
      "country_code": "US",
      "zip": "21037",
      "phone_number": "+17285191522",
      "email": "Ezra.Dorris@barmail.com",
      "name": "Ezra Dorris",
      "order_status": "Shipped",
      "payment_ref": "yeFMfAZo",
      "transaction_id": "EjP1AfEm4Gw3",
      "payment_amt_cents": 3282,
      "ship_charged_cents": 1310,
      "ship_cost_cents": 728,
      "subtotal_cents": 4592,
      "total_cents": 4592,
      "shipper_name": "FedEx",
      "payment_date": "2018-06-17T18:44:59Z",
      "shipped_date": "2018-09-09T05:19:58Z",
      "tracking_number": "H5411491623PR",
      "tax_total_cents": 127,
      "created_at": "2018-02-17T02:04:52Z",
      "updated_at": "2018-12-18T10:37:33Z"
    },
    {
      "id": 109919,
      "product_id": 912,
      "inventory_id": 5280,
      "street_address": "7111 North 75th Avenue",
      "apartment": "#1067",
      "city": "Glendale",
      "state": "AZ",
      "country_code": "US",
      "zip": "85303",
      "phone_number": "+18141733209",
      "email": "Hertha.Urena@blargmail.org",
      "name": "Hertha Urena",
      "order_status": "Shipped",
      "payment_ref": "XV14VJ9k",
      "transaction_id": "jTRe1PPUhp2N",
      "payment_amt_cents": 2626,
      "ship_charged_cents": 1296,
      "ship_cost_cents": 1208,
      "subtotal_cents": 3922,
      "total_cents": 3922,
      "shipper_name": "DHL",
      "payment_date": "2018-02-16T06:23:15Z",
      "shipped_date": "2018-03-04T23:18:19Z",
      "tracking_number": "n9499421954bD",
      "tax_total_cents": 257,
      "created_at": "2018-01-29T12:15:02Z",
      "updated_at": "2018-06-21T18:15:48Z"
    }
  ],
  "total": 1196
}
```

Order Status Options

The Order status options endpoint will return an array of the status string values which can be used to filter the order set.

GET `https://cscodetest.herokuapp.com/api/orders/options`

Response:

```
Response Code 200
{
  "order_status_options": [
    "fulfulled",
    "open",
    "paid",
    "pending",
    "shipped"
  ]
}
```

Order Item

The order endpoint will return an individual order record based on the `:id` parameter is provided.

GET `https://cscodetest.herokuapp.com/api/order/:id`

Response:

The response will be the order payload for the order record with the associated id.

```
Response Code 200
{
  "order": {
    "id": 109919,
    "product_id": 912,
    "inventory_id": 5280,
    "street_address": "7111 North 75th Avenue",
    "apartment": "#1067",
    "city": "Glendale",
    "state": "AZ",
    "country_code": "US",
    "zip": "85303",
    "phone_number": "+18141733209",
    "email": "Hertha.Urena@blargmail.org",
    "name": "Hertha Urena",
    "order_status": "Shipped",
    "payment_ref": "XV14VJ9k",
    "transaction_id": "jTRe1PPUhp2N",
    "payment_amt_cents": 2626,
    "ship_charged_cents": 1296,
    "ship_cost_cents": 1208,
    "subtotal_cents": 3922,
    "total_cents": 3922,
    "shipper_name": "DHL",
    "payment_date": "2018-02-16T06:23:15Z",
    "shipped_date": "2018-03-04T23:18:19Z",
    "tracking_number": "n9499421954bD",
    "tax_total_cents": 257,
    "created_at": "2018-01-29T12:15:02Z",
    "updated_at": "2018-06-21T18:15:48Z"
  }
}
```

Order Create

Order create will create a new order record for the associated user account. To create a new order you should specify valid order and inventory id values.

POST `https://cscodetest.herokuapp.com/api/order`

Payload:

```
{
    "product_id": 7499,
    "inventory_id": 18543,
    "street_address": "123 Fake St",
    "apartment": "",
    "city": "New York",
    "state": "NY",
    "country_code": "US",
    "zip": "10011",
    "phone_number": "+34743562191",
    "email": "some.guy@guy.com",
    "name": "Phil Guy",
    "payment_amt_cents": 1499,
    "tax_total_cents": 123,
    "subtotal_cents": 1622,
    "total_cents": 1622
}
```

Response:

The response includes a success message along with the id of the created order record.

```
Response Code 200
{
  "message": "Created successfully",
  "order_id": 110729
}
```

Order Update

When updating an order you can provide any subset of the parameters used to create a new record. Specify the `:id` of the order to be updated in the url.

PUT `https://cscodetest.herokuapp.com/api/order/:id`

Example payload:

```
{

    "street_address": "123 Fake Ave",
    "apartment": "Apt 5C"
}
```

Response:

Response includes success message and the order id of the updated record.

```
Response Code 200
{
    "message": "Updated successfully",
   "order_id": 110729
}
```

Order Delete

The order delete endpoint allows for deletion of records that the user owns (they are tied to products under the user account). Make a `DELETE` call to the url and set the `:id` to the record id that should be deleted.

DELETE `https://cscodetest.herokuapp.com/api/order/:id`

Response:

Response includes success message and the id of the order that was deleted.

```
{
  "message": "Deleted successfully",
  "order_id": 110729
}
```

Valid User Accounts

`betty.iraheta@foomail.org` , `IwBbzQUzcD`

`bradly.rouillard@rigmail.com` , `TSXsfZxMls`

`maxima.legler@rigmail.com` , `dgDjsoAXWM`

`providencia.toombs@foo.com` , `hGuVoyBrkG`

`hortensia.dollison@blargmail.org` , `AgcGcJxeig`

`tamar.poyer@barmail.com` , `hdSEjsQWxg`

`truman.marcos@foomail.org` , `AqSMUhGxgy`

`michelina.corvin@foo.com` , `hRVBnoRbaj`