Brewbean API Documentation

Brewbean APIs

Version: v1.0.0

Update Date: March 11, 2022

Overview

Brewbean-backend allows users to develop applications that integrate with a customer instance of the brewbean coffee shop. The APIs are Rest services developed using Node.js and allow users/developers to retrieve and store data for the online coffee shop.

Brewbean APIs

Host the data of the Brewbean Coffee Shop website.

There are multiple endpoints/API actions available for this API, and this document describes them in detail:

Users API Action #1

Functionality	Get the users of the Brewbean website application.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/users
HTTP Methods	GET
Scope	This API allows a user to get the users account information that was supplied on the Brewbean website.

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
isAdmin	Boolean	A boolean value that gives the user admin rights as a user in the system.
id	String	A unique string value that identifies each user.
email	String	A string value that represents the users email address in the system.
firstname	String	A string value that represents the user's firstname in the system.
lastname	String	A string value that represents the user's lastname in the system.
password	String	An encrypted string value that represents the user's password in the system.

Sample Response:

```
{
    "isAdmin": false,
    "_id": "60b8570aeb475b3f2084a9c6",
    "email": "test2@gmail.com",
    "firstname": "Jane",
    "lastname": "Doe",
    "password": "$2b$10$vKc/EUWszC4hhxNxej7UPuDpWx6WBPCkLA13PCO7bHrxHi0SX5mi2",
    "__v": 0
},
{
    "isAdmin": true,
    "_id": "60b8569ceb475b3f2084a9c5",
    "email": "test1@gmail.com",
    "firstname": "John",
    "lastname": "Doe",
    "password": "$2b$10$wkCAdLBY8Ocn5Xb66S0.NugcEVSF8ek2LzijbXwanwsnOrEpvxXGq",
    "__v": 0
}
```

Users API Action #2

Functionality	Get the current user based on userid from the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/users/:id
HTTP Methods	GET
Scope	This API allows a user to get specific user account information that was supplied in the system.

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
isAdmin	Boolean	A boolean value that gives the user admin rights as a user in the system.
id	String	A unique string value that identifies each user.
email	String	A string value that represents the users email address in the system.
firstname	String	A string value that represents the user's firstname in the system.
lastname	String	A string value that represents the user's lastname in the system.
password	String	The user's password will be encrypted (bcrypt) before being stored in the database.

Sample Response:

```
{
    "isAdmin": false,
    "_id": "60b8570aeb475b3f2084a9c6",
    "email": "test2@gmail.com",
    "firstname": "Jane",
    "lastname": "Doe",
    "password": "$2b$10$vKc/EUWszC4hhxNxej7UPuDpWx6WBPCkLA13PCO7bHrxHi0SX5mi2",
    "__v": 0
}
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 404 Not Found

Invalid ID.

Users API Action #3

Functionality	Add the user to the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/users/
HTTP Methods	POST
Scope	This API allows a user's information to be stored in a MongoDB database on the backend.

Request Body Parameters for API Action #3

Input Param	Datatype	REQ*	Description
email	String	Required	A valid email address for a user placing an order on the website. Must be a unique value.
firstname	String	Required	User's firstname. Must be 3 or more characters in length.
lastname	String	Required	User's lastname. Must be 3 or more characters in length.
password	String	Required	User's password. Must be 5 or more characters in length.
isAdmin	Boolean	Optional	Set user's rights to Administrator. Default value is false.

Sample Request:

```
{
    "email": "test3@gmail.com",
    "firstname": "Tom",
    "lastname": "Cat",
    "password": "3password",
    "isAdmin": false
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	Boolean	A boolean value that gives the user admin rights as a user in the system.
email	String	A unique string value that identifies each user.
firstname	String	A string value that represents the user's firstname.
lastname	String	A string value that represents the user's lastname.
isAdmin	String	An encrypted string value that represents the user's password.

Sample Response:

```json
{
    "_id": "60b94e5cbdcdac0c14470ebe",
    "email": "test3@gmail.com",
    "firstname": "Tom",
    "lastname": "Cat",
    "isAdmin": false
}
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

Something failed. Error has been logged.

Users API Action #4

Functionality	Update the user's account information in the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/users/:id
HTTP Methods	PUT
Scope	This API allows a user's information to be updated in a MongoDB database on the backend. To use this API user completing update must have Admin rights (isAdmin=true).

Request Body Parameters for API Action #4

Input Param	Datatype	REQ*	Description
email	String	Required	Update email address for a user placing an order on the website. Must be a unique value. Enter current value if value will not change.
firstname	String	Required	Update user's firstname. Must be 3 or more characters in length. Enter current value if value will not change.
lastname	String	Required	Update user's lastname. Must be 3 or more characters in length. Enter current value if value will not change.
password	String	Required	User's password. Must be 5 or more characters in length. Enter current value if value will not change.
isAdmin	Boolean	Optional	Update user Administrator rights.

Sample Request:

```
{
    "email": "test3update@gmail.com",
    "firstname": "Tommy",
    "lastname": "Cat",
    "password": "3passwordUpdate",
    "isAdmin": false
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
isAdmin	Boolean	A boolean value that gives the user admin rights as a user in the system.
id	String	A unique string value that identifies each user.
email	String	A string value that represents the user's updated email.
firstname	String	A string value that represents the user's updated firstname
lastname	String	A string value that represents the user's updated lastname.
password	String	A string value that represents the user's updated password.

Sample Response:

```
{
    "isAdmin": false,
    "_id": "60b94e5cbdcdac0c14470ebe",
    "email": "test3update@gmail.com",
    "firstname": "Tommy",
    "lastname": "Cat",
    "password": "3passwordUpdate",
    "__v": 0
}
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

"password" is required

Or

Something failed. Error has been logged.

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

404 Not Found

Invalid ID.

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Review the table below to understand this API's 403 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 403 Forbidden

Access is denied.

Users API Action #5

Functionality	Delete the user from the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/users/:id
HTTP Methods	DELETE
Scope	This API allows a user to be deleted from the MongoDB database on the backend. To use this API user completing update must have Admin rights (isAdmin=true).

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
isAdmin	Boolean	A boolean value that gives the user admin rights as a user in the system.
id	String	A unique string value that identifies the user that's deleted from the system.
email	String	A string value that represents the user's email address that will be deleted.

firstname	String	A string value that represents the user's firstname that will be deleted.
lastname	String	A string value that represents the user's lastname that will be deleted.
password	String	The user's password that will be deleted from the system.

Sample Response:

```
"isAdmin": false,
"_id": "60b94e5cbdcdac0c14470ebe",
"email": "test3update@gmail.com",
"firstname": "Tommy",
"lastname": "Cat",
"password": "3passwordUpdate",
"__v": 0
```

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Review the table below to understand this API's 403 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:
Status: 403 Forbidden

Access is denied.

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Status: 404 Not Found

Invalid ID.

Authentication API Action #1

Functionality	Validate the user logging into the Brewbean website application.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/authentication
HTTP Methods	POST
Scope	This API validates the user logging into the Brewbean website. If they enter a valid username and password a JSON web token is created and stored in the x-access-token header for each http request that requires a token to authenticate.

Request Body Parameters for API Action #1

Input Param	Datatype	REQ*	Description
email	String	Required	A valid email address for a user logging into the system.
password	String	Required	User's password. Must be 5 or more characters in length.

Sample Request:

```
{
    "email": "test4@gmail.com",
    "password": "password4"
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
auth	Boolean	A Boolean value that returns true if the user authentication passes with a valid email, password and token. It returns false if the authentication fails.
token	String	A string value that represents the JSON web token for authenticating http requests.

Sample Response:
```
{
    "auth": true,
    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MjI4MzI0MjljMjYzODAwMT
Y3OGU3ZDkiLCJpYXQiOjE2NDY4ODU2MDgsImV4cCI6MTY0Njk3MjAwOH0.Ab5rD
MY4TYMFobOOFHTVa0Go8T4gjtVAfj1IFMT2E3k"
}
```

Coffee API Action #1

Functionality	Get the coffee items from the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/coffee
HTTP Methods	GET
Scope	This API allows a user to retrieve all available coffee items from the system stored in a MongoDB database. This API requires an x-access-token key header in the request that contains a JSON web token.

☑ x-access-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO
WY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz
MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP
hLwCC47cutevz5h6uv5IpdC8

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
coffee	Array	A list of available coffees in the system.
creamer	Array	A list of available coffee creamers in the system.
topping	Array	A list of available coffee toppings in the system.
syrup	Array	A list of available coffee syrups in the system.
sweetener	Array	A list of available coffee sweeteners in the system.
price	Number	The price of the coffee in the system.
id	String	Unique id generated from MongoDB database.

Sample Response:

```json
  "coffee": [
      "blonde roast",
      "french roast",
      "dark roast",
      "columbia",
      "breakfast blend"
  ],
  "creamer": [
      "no creamer",
      "french vanilla",
      "half&half",
      "hazelnut"
  ],
  "topping": [
      "no topping",
      "whipcream",
      "caramel drizzle",
      "mocha drizzel"
  ],
  "syrup": [
      "no syrup",
      "hazelnut syrup",
      "irish cream syrup",
      "peppermint syrup"
  ],
  "sweetener": [
      "no sugar",
      "sugar",
      "stevia",
      "splenda"
  ],
  "_id": "60bd5011b737163f9803a7fc",
  "price": 4.5,
  "__v": 0
```

Coffee API Action #2

Functionality	Get the current coffee items based on id.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/coffee/:id
HTTP Methods	GET
Scope	This API allows a user to get specific coffee items stored in the system. This API requires an x-access-token key header in the request that contains a JSON web token.

☑ x-access-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO
WY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz
MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP
hLwCC47cutevz5h6uv5IpdC8

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
coffee	Array	A list of available coffees in the system.
creamer	Array	A list of available coffee creamers in the system.
topping	Array	A list of available coffee toppings in the system.
syrup	Array	A list of available coffee syrups in the system.
sweetener	Array	A list of available coffee sweeteners in the system.
price	Number	The price of the coffee in the system.
id	String	Unique Id generated from MongoDB database.

Sample Response:

```
"coffee": [
      "blonde roast",
      "french roast",
      "dark roast",
      "columbia",
      "breakfast blend"
],
"creamer": [
      "no creamer",
      "french vanilla",
      "half&half",
      "hazelnut"
],
"topping": [
      "no topping",
      "whipcream",
      "caramel drizzle",
      "mocha drizzel"
],
"syrup": [
      "no syrup",
      "hazelnut syrup",
      "irish cream syrup",
      "peppermint syrup"
],
"sweetener": [
      "no sugar",
      "sugar",
      "stevia",
      "splenda"
],
"_id": "60bd5011b737163f9803a7fc",
"price": 4.5,
"__v": 0
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 404 Not Found

Invalid ID.

Coffee API Action #3

Functionality	Add the coffee items to the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/coffee/
HTTP Methods	POST
Scope	This API allows coffee items to be stored in a MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token.

☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO WY5OGVlM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP hLwCC47cutevz5h6uv5IpdC8
☐		

Request Body Parameters for API Action #3

Input Param	Datatype	REQ*	Description
coffee	Array	Required	An array of at least three coffee types.
creamer	Array	Required	An array of at least three creamers for the coffee types.
topping	Array	Required	An array of at least three toppings for the coffee types.
syrup	Array	Required	An array of at least three syrups for the coffee types.
sweetener	Array	Required	An array of at least three sweeteners for the coffee types.
price	Price	Required	The total price of the coffee including its additional items (creamer, topping, etc.)

Sample Request:

```
{
    "coffee": ["blonde roast", "french roast", "dark roast", "columbia", "breakfast blend"],
    "creamer": ["no creamer", "french vanilla", "half&half", "hazelnut"],
    "topping": ["no topping", "whipcream", "caramel drizzle", "mocha drizzel"],
    "syrup": ["no syrup", "hazelnut syrup", "irish cream syrup", "peppermint syrup"],
    "sweetener": ["no sugar", "sugar", "stevia", "splenda"],
    "price": 4.50
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
coffee	Array	A list of available coffees in the system.
creamer	Array	A list of available coffee creamers in the system.
topping	Array	A list of available coffee toppings in the system.
syrup	Array	A list of available coffee syrups in the system.
sweetener	Array	A list of available coffee sweeteners in the system.
price	Number	The price of the coffee in the system.
id	String	Unique id generated from MongoDb database.

Sample Response:

```json
"coffee": [
    "blonde roast",
    "french roast",
    "dark roast",
    "columbia",
    "breakfast blend"
],
"creamer": [
    "no creamer",
    "french vanilla",
    "half&half",
    "hazelnut"
],
"topping": [
    "no topping",
    "whipcream",
    "caramel drizzle",
    "mocha drizzel"
],
"syrup": [
    "no syrup",
    "hazelnut syrup",
    "irish cream syrup",
    "peppermint syrup"
],
"sweetener": [
    "no sugar",
    "sugar",
    "stevia",
    "splenda"
],
"_id": "60bd5011b737163f9803a7fc",
"price": 4.5,
"__v": 0
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

Something failed. Error has been logged.

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Coffee API Action #4

Functionality	Update the coffee items in the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/coffee/:id
HTTP Methods	PUT
Scope	This API allows a coffee item to be updated in a MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token.

x-access-token   eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO
WY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz
MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP
hLwCC47cutevz5h6uv5IpdC8

Request Body Parameters for API Action #4

Input Param	Datatype	REQ*	Description
coffee	Array	Required	Update coffee. Must be 3 or more elements in the array and a unique value. Enter current values if values will not change.
creamer	Array	Required	Update creamer. Must be 3 or more elements in the array. Enter current values if values will not change.
topping	Array	Required	Update topping. Must be 3 or more elements in the array. Enter current values if values will not change.
syrup	Array	Required	Update syrup. Must be 3 or more elements in the array. Enter current values if values will not change.
sweetener	Array	Required	Update sweetener. Must be 3 or more elements in the array. Enter current values if values will not change.
price	Number	Required	Update price. Enter current value if value will not change.

Sample Request:

```
{
    "coffee": ["blonde roast", "french roast", "dark roast", "columbia", "breakfast blend"],
    "creamer": ["no creamer", "french vanilla", "half&half", "hazelnut"],
    "topping": ["no topping", "whipcream", "caramel drizzle", "mocha drizzel"],
    "syrup": ["no syrup", "hazelnut syrup", "irish cream syrup", "peppermint syrup"],
    "sweetener": ["no sugar", "sugar", "stevia", "splenda"],
    "price": 5.50
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
coffee	Array	A list of available coffees in the system.
creamer	Array	A list of available coffee creamers in the system.
topping	Array	A list of available coffee toppings in the system.
syrup	Array	A list of available coffee syrups in the system.
sweetener	Array	A list of available coffee sweeteners in the system.
price	Number	The price of the coffee in the system.
id	String	Unique id generated from MongoDb database.

Sample Response:

```json
{
    "coffee": [
        "blonde roast",
        "french roast",
        "dark roast",
        "columbia",
        "breakfast blend"
    ],
    "creamer": [
        "no creamer",
        "french vanilla",
        "half&half",
        "hazelnut"
    ],
    "topping": [
        "no topping",
        "whipcream",
        "caramel drizzle",
        "mocha drizzel"
    ],
    "syrup": [
        "no syrup",
        "hazelnut syrup",
        "irish cream syrup",
        "peppermint syrup"
    ],
    "sweetener": [
        "no sugar",
        "sugar",
        "stevia",
        "splenda"
    ],
    "_id": "60bd5011b737163f9803a7fc",
    "price": 4.5,
    "__v": 0
}
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

"topping" is required

Or

Something failed. Error has been logged.

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

404 Not Found

Invalid ID.

Coffee API Action #5

Functionality	Delete the coffee items from the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/coffee/:id
HTTP Methods	DELETE
Scope	This API allows coffee items to be deleted from the MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token. To use this API user completing update must have Admin rights (isAdmin=true).

☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkOWY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjPhLwCC47cutevz5h6uv5IpdC8
☐		

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
coffee	Array	The array of coffees that were deleted from the system.
creamer	Array	The array of creamers that were deleted from the system.
topping	Array	The array of toppings that were deleted from the system.
syrup	Array	The array of syrups that were deleted from the system.
sweetener	Array	The array of sweeteners that were deleted from the system.
price	Number	The coffee price that was deleted from the system.
id	String	The coffee id deleted from the system.

Sample Response:

```
{
    "coffee": [
        "blonde roast",
        "french roast",
        "dark roast",
        "columbia",
        "breakfast blend"
    ],
    "creamer": [
        "no creamer",
        "french vanilla",
        "half&half",
        "hazelnut"
    ],
    "topping": [
        "no topping",
        "whipcream",
        "caramel drizzle",
        "mocha drizzel"
    ],
    "syrup": [
        "no syrup",
        "hazelnut syrup",
        "irish cream syrup",
        "peppermint syrup"
    ],
    "sweetener": [
        "no sugar",
        "sugar",
        "stevia",
        "splenda"
    ],
    "_id": "60bd5011b737163f9803a7fc",
    "price": 4.5,
    "__v": 0
}
```

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Review the table below to understand this API's 403 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 403 Forbidden

```
Access is denied.
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Status: 404 Not Found

```
Invalid ID.
```

Orders API Action #1

Functionality	Get the current orders in the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders
HTTP Methods	GET
Scope	This API retrieves all orders in the system. This API requires an x-access-token key header in the request that contains a JSON web token.

| ☑ | x-access-token | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkOWY5OGVlM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMzMzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjPhLwCC47cutevz5h6uv5lpdC8 |
| ☐ | | |

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	String	Unique id generated from MongoDB database for order.
coffee	String	The coffee ordered.
creamer	String	The creamer ordered.
topping	String	The topping ordered.
syrup	String	The syrup ordered.
sweetener	String	The sweetener ordered.
price	Number	The price of the coffee ordered.
quantity	Number	The amount of coffees ordered.
subtotal	Number	The subtotal of the order (price * quantity)
User	String	The user's id that placed the order.

Sample Response:

```json
{
        "_id": "60c00c74ef30535dd47f1477",
        "coffee": "columbia",
        "creamer": "french vanilla",
        "topping": "no topping",
        "syrup": "no syrup",
        "sweetener": "stevia",
        "price": 4.5,
        "quantity": 2,
        "subTotal": 9,
        "user": "60bfdd7f4550785318a0d691",
        "__v": 0
},
{
        "_id": "60c016ba25fcc9600cc060aa",
        "coffee": "dark roast",
        "creamer": "half & half",
        "topping": "no topping",
        "syrup": "no syrup",
        "sweetener": "sugar",
        "price": 4.5,
        "quantity": 1,
        "subTotal": 4.5,
        "user": "60bfde314550785318a0d692",
        "__v": 0
}
```

Orders API Action #2

Functionality	Get the order based on order id.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders/:id
HTTP Methods	GET
Scope	This API allows a user to get specific order from the system based on order id. This API requires an x-access-token key header in the request that contains a JSON web token.

	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO WY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP hLwCC47cutevz5h6uv5lpdC8

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	String	Id used in endpoint to return the order based on this id.
coffee	String	Coffee ordered based on order id.
creamer	String	Creamer ordered based on order id.
topping	String	Topping ordered based on order id.
syrup	String	Syrup ordered based on order id.
sweetener	String	Sweetener ordered based on order id.
price	Number	The price of the coffee ordered.
quantity	Number	Amount of coffees ordered.
subtotal	Number	The subtotal of the order (price * quantity).
user	String	The user's id that placed the order.

Sample Response:

```
{
    "_id": "60c016ba25fcc9600cc060aa",
    "coffee": "dark roast",
    "creamer": "half & half",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "sugar",
    "price": 4.5,
    "quantity": 1,
    "subTotal": 4.5,
    "user": "60bfde314550785318a0d692",
    "__v": 0
}
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 404 Not Found

Invalid ID.

Orders API Action #3

Functionality	Get the user's order based on user id.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders/getOrder
HTTP Methods	GET
Scope	This API allows a user to get specific order from the system based on the user's id. This API requires an x-access-token key header in the request that contains a JSON web token.

☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkOWY5OGVlM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMzMzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjPhLwCC47cutevz5h6uv5IpdC8
☐		

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	String	Order id of the order placed.
coffee	String	Coffee ordered by user.
creamer	String	Creamer ordered by user.
topping	String	Topping ordered by user.
syrup	String	Syrup ordered by user.
sweetener	String	Sweetener ordered by user.
price	Number	The price of the coffee ordered.
quantity	Number	Amount of coffees ordered.
subtotal	Number	The subtotal of the order (price * quantity).
user	String	The user's id that placed the order.

Sample Response:

```
{
    "_id": "60c016ba25fcc9600cc060aa",
    "coffee": "dark roast",
    "creamer": "half & half",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "sugar",
    "price": 4.5,
    "quantity": 1,
    "subTotal": 4.5,
    "user": "60bfde314550785318a0d692",
    "__v": 0
}
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 404 Not Found

An order could not be found for this user

Orders API Action #4

Functionality	Add the order to the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders/
HTTP Methods	POST
Scope	This API allows order to be stored in a MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token.

x-access-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO
WY5OGVIM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz
MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP
hLwCC47cutevz5h6uv5IpdC8

Request Body Parameters for API Action #4

Input Param	Datatype	REQ*	Description
coffee	String	Required	A string of at least five characters.
creamer	String	Required	A string of at least five characters.
topping	String	Required	A string of at least five characters.
syrup	String	Required	A string of at least five characters.
sweetener	String	Required	A string of at least five characters.
price	Price	Required	A number representing the price of the coffee
quantity	Number	Required	A number representing the amount of coffees ordered.
user	String	Required	User id from user schema that the order is for.

Sample Request:

```
{
    "coffee": "columbia",
    "creamer": "french vanilla",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "stevia",
    "price": 4.50,
    "quantity": 2,
    "user": "60bfdd7f4550785318a0d691"

}
```

Output Param	Datatype	Description
Id	String	Unique id generated from MongoDb database for the order.
coffee	String	Name of coffee ordered.
creamer	String	Name of creamer ordered.
topping	String	Name of topping ordered.
syrup	String	Name of syrup ordered.
sweetener	String	Name of sweetener ordered.
price	Number	The price of the coffee ordered.
quantity	Number	The quantity of the coffee ordered.
subtotal	Number	Subtotal of ordered coffee.
user	String	Id of the user the order is for.

Sample Response:

```
{
    "_id": "60c00c74ef30535dd47f1477",
    "coffee": "columbia",
    "creamer": "french vanilla",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "stevia",
    "price": 4.5,
    "quantity": 2,
    "subTotal": 9,
    "user": "60bfdd7f4550785318a0d691",
    "__v": 0
}
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

Please enter a quantity greater than zero.

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

Please enter a quantity greater than zero.

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Orders API Action #5

Functionality	Update the order in the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders/:id
HTTP Methods	PUT
Scope	This API allows an order be updated in a MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token.

x-access-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkO
WY5OGVlM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMz
MzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjP
hLwCC47cutevz5h6uv5IpdC8

Request Body Parameters for API Action #5

Input Param	Datatype	REQ*	Description
coffee	String	Required	Update coffee. Must be 5 or more characters long. Enter current value if value will not change.
creamer	String	Required	Update creamer. Must be 5 or more characters long. Enter current value if value will not change.
topping	String	Required	Update topping. Must be 5 or more characters long. Enter current value if value will not change.
syrup	String	Required	Update syrup. Must be 5 or more characters long. Enter current value if value will not change.
sweetener	String	Required	Update sweetener. Must be 5 or more characters long. Enter current value if value will not change.
price	Number	Required	Update price. Enter current value if value will not change.
User	String	Required	Update user id. Enter current value if value will not change.

Sample Request:

```json
{
    "coffee": "dark roast",
    "creamer": "half & half",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "sugar",
    "price": 4.50,
    "quantity": 2,
    "user": "60bfdd7f4550785318a0d691"
}
```

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	String	Id of order that's been updated.
coffee	Array	Coffee name that may have been updated.
creamer	Array	Creamer name that may have been updated.
topping	Array	Topping name that may have been updated.
syrup	Array	Syrup name that may have been updated.
sweetener	Array	Sweetener name that may have been updated.
price	Number	Price of order that may have been updated.
quantity	Number	Quantity of coffees that may have been updated.
subtotal	Number	Subtotal of updated order.
user	String	User id that may have been updated in the order.

Sample Response:

```json
{
    "_id": "60c00c74ef30535dd47f1477",
    "coffee": "columbia",
    "creamer": "french vanilla",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "stevia",
    "price": 4.5,
    "quantity": 2,
    "subTotal": 9,
    "user": "60bfdd7f4550785318a0d691",
    "__v": 0
}
```

Review the table below to understand this API's 400 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 400 Bad Request

```
"topping" is required
```

Or

```
Something failed. Error has been logged.
```

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 404 Not Found

Invalid ID.

Orders API Action #6

Functionality	Delete the order from the system.
Base Path	https://warm-dusk-21815.herokuapp.com/
Dependency	None
Endpoint URL / Resource	/api/v1/orders/:id
HTTP Methods	DELETE
Scope	This API allows orders to be deleted from the MongoDB database on the backend. This API requires an x-access-token key header in the request that contains a JSON web token. To use this API user completing update must have Admin rights (isAdmin=true).

☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjZkOWY5OGVlM2U0MDAxNjY1YWRhMyIsImlhdCI6MTY0NjcxMzMzNywiZXhwIjoxNjQ2Nzk5NzM3fQ.Z8QvtZVKVR0mw1DjvjPhLwCC47cutevz5h6uv5IpdC8
☐		

Review the table below to understand this API's 200 response parameters:

Output Param	Datatype	Description
id	String	
coffee	Array	The array of coffees that were deleted from the system.
creamer	Array	The array of creamers that were deleted from the system.
topping	Array	The array of toppings that were deleted from the system.
syrup	Array	The array of syrups that were deleted from the system.
sweetener	Array	The array of sweeteners that were deleted from the system.
price	Number	The coffee price that was deleted from the system.
quantity	Number	The coffee id deleted from the system.
subtotal	Number	
user	String	

Sample Response:

```
{
    "_id": "60c00c74ef30535dd47f1477",
    "coffee": "dark roast",
    "creamer": "half & half",
    "topping": "no topping",
    "syrup": "no syrup",
    "sweetener": "sugar",
    "price": 4.5,
    "quantity": 2,
    "subTotal": 9,
    "user": "60bfde314550785318a0d692",
    "__v": 0
}
```

Review the table below to understand this API's 401 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 401 Unauthorized

Access Denied. No token provided.

Review the table below to understand this API's 403 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Sample Response:

Status: 403 Forbidden

Access is denied.

Review the table below to understand this API's 404 response parameters:

Output Param	Datatype	Description
code	String	The error code.
message	String	Textual description of the problem.

Status: 404 Not Found

Invalid ID.