Emily Yu

Boba Shop: API Specification

CS 493: Cloud Application Development

Fall 2022

Oregon State University

Last Update: November 27, 2022

URL

https://cs493-finalproject-369522.wl.r.appspot.com

Change log

Version	Change	Date
1.0	Initial version.	Nov 27, 2022

Data Model

The app stores three entities in Datastore: Users, Drinks, and Orders.

Users

Property	Data Type	Notes
user_id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
user_sub	String	User's sub
name	String	User's name
email	string	User's email
order	Integer	Orders associated with patron
self	String	URL to the user

Drinks

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Name of drink
tea	String	Name of tea
topping	String	Toppings in drink
order_id	Integer	Order id corresponding with drink
self	String	URL to the drink

Orders

Property	Data Type	Notes
order_id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
date	String	string
customer_information	Integer	User sub corresponding to the order
size	String	Size of drink
drink	integer	Drink id corresponding to the order
self	String	URL to the boat.

Drinks

Create a Drink

Allows you to create a new drink.

POST /drinks

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the drink.	Yes
tea	Type of tea	Yes
topping	Type of topping	Yes

Request Body Example

```
{
   "name": "BMT",
   "tea": "Black Tea",
   "topping": “boba”
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don’t need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute’s value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	403 Forbidden	User input name of drink is not unique.

Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 201 Created (JSON)
{
        "id": 123,
       "tea": "BMT",
        "topping": “Boba”,
       “order”: []
        "self": http://localhost:8000/drinks/123
}
```

Failure

```
Status: 400 Bad Request
{
   "Error":  "Bad Request. The request object is missing at least one of the required attributes"
}

Status: 403
{
   “Error" : "Forbidden. The request object ['name'] is not unique."
}
{
   "Error" : "Forbidden. The request object has invalid data type."
}

Status: 406 Not Acceptable
{
   "Error": "Not Acceptable. MIME type is not supported."
}
```

List all Drinks

List all the drinks.

GET /drinks

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns either HTML or JSON depending on accept header
Error	415 Unsupported Media Type	MIME type is not supported

Response Examples

Success

Status: 200 OK (JSON)

```
[
{   "id": 123,
   "name": "BMT",
   "tea": "Black Tea",
   "topping": “boba”
   "order": []
   "self": http://localhost:8000/drinks/123
},
{   "id": 456,
   "name": "JMT w/ Boba",
   "tea": "Jasmine Milk Tea",
   "topping": “boba”
   "order": []
   "self": http://localhost:8000/drinks/456
}
]
```

Status: 200 OK (HTML)

```
<html>
<head></head>
<body>
  <ul>
```

```
    <li>Boat ID: 123</li>
    <ul>
      <li>Name: BMT </li>
      <li>Tea: Black Tea </li>
      <li>Topping: Boba</li>
      <li>Order: []</li>
      <li>Self: http://localhost:8080/drinks/123</li>
    </ul>
    <li>Boat ID: 456</li>
    <ul>
      <li>Name: JMT w/ Boba </li>
      <li>Tea: Jasmine Milk Tea </li>
      <li>Topping: Boba</li>
      <li>Order: []</li>
      <li>Self: http://localhost:8080/drinks/456</li>
    </ul>
</body>
</html>
```

Failure

```
Status: 415 Unsupported Media Type
{
   "Error": "Unsupported Media Type. Please submit 'application/json' or 'text/html' MIME type."
}
```

DELETE/PUT/PATCH a Drink

Allows you to get an existing boat.

[DELETE/PUT/PATCH] /drinks

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	Response is not supported

Response Examples

Success

N/A

Failure

```
Status: 405 Not Found
{
   "Error": "Method Not Allowed. Allowed Methods: GET, POST."
}
```

GET a Drink

Allows you to get an existing drink.

GET /drinks/:drink_id

Request

Path Parameters

Name	Description
drink_id	ID of the drink

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No drink with this drink_id exists
Failure	415 Unsupported Media Type	Media type is not supported

Response Examples

Success

```
Status: 200 OK (JSON)
{
   "id": 123,
   "name": "BMT",
   "tea": "Black Tea",
   "topping": “boba”
   "order": []
   "self": http://localhost:8000/drinks/123
}
```

Failure

```
Status: 404 Not Found
{
   "Error": "Not Found. No drink with this drink_id exists."
}
Status: 415 Unsupported Media Type
{
   "Error":  " Unsupported Media Type. Please submit 'application/json' or 'text/html' MIME type."
}
```

Edit a Drink

Allows you to edit a drink.

PUT /drinks

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the drink.	Yes
type	Type of tea	Yes
topping	Type of topping	Yes

Request Body Example

```
{
   "name": "Black Milk Tea",
   "tea": "Black Tea",
   "topping": “boba”
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don’t need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute’s value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	403 Forbidden	User input name of drink is not unique.
Failure	404 Not Found	No drink with this drink_id exists
Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

Status: 200 OK (JSON)

```
{
       "id": 123,
       "name": "Black Milk Tea",
       "tea": "Black Tea",
       "topping": “Boba”,
      “order”: []
       "self": http://localhost:8000/drinks/123
}
```

Failure

Status: 400 Bad Request

```
{
  "Error":  "Bad Request. The request object is missing at least one of the required attributes"
}
```

Status: 403

```
{
   “Error" : "Forbidden. The request object ['name'] is not unique."
}
{
  "Error" : "Forbidden. The request object has invalid data type."
}
```

Status: 404 Not Found

```
{
   "Error": "Not Found. No drink with this drink_id exists."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. MIME type is not supported."
}
```

Patch a Drink

Allows you to edit a drink.

PATCH /drinks

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the drink.	No
type	Type of tea	No
topping	Type of topping	No

Request Body Example

```
{
   "name": "Black Milk Tea",
   "tea": "Black Tea",
   "topping": “boba”
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don’t need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute’s value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	403 Forbidden	User input name of drink is not unique.
Failure	404 Not Found	No drink with this drink_id exists

Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

Status: 200 OK (JSON)

```
{
       "id": 123,
       "name": "Black Milk Tea",
       "tea": "Black Tea",
       "topping": “Boba”,
      “order”: []
       "self": http://localhost:8000/drinks/123
}
```

Failure

Status: 400 Bad Request

```
{
  "Error":  "Bad Request. The request object is missing at least one of the required attributes"
}
```

Status: 403

```
{
   “Error" : "Forbidden. The request object ['name'] is not unique."
}
{
  "Error" : "Forbidden. The request object has invalid data type."
}
```

Status: 404 Not Found

```
{
   "Error": "Not Found. No drink with this drink_id exists."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable. MIME type is not supported."
}
```

DELETE a Drink

Allows you to delete an existing drink.

DELETE /drinks/:drink_id

Request

Path Parameters

Name	Description
drink_id	ID of the drink

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	404 Not Found	No drink with this drink_id exists

Response Examples

Success

Status: 204 OK (JSON)

Failure

```
Status: 404 Not Found
{
   "Error": "Not Found. No drink with this drink_id exists."
}
```

Orders

Create an Order

Allows you to create a new order.

POST /orders

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
date	Date	Yes
size	Size of the drink	Yes
drink_id	drink id	Yes

Request Body Example

```
{
   "date": "05-05-05",
   "size": "Medium",
   "drink_id": 5081054809423872
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).

Failure	401 Unauthorized	JWT not verified
Failure	403 Forbidden	Data type is incorrect.
Failure	404 Not Found	Drink ID does not exist.
Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 201 Created (JSON)
{
  "date": "05-05-05",
  "drink": 5081054809423872,
  "order_id": 5677459739508736,
  "owner": "auth0|63858def67ad3e09030cd3fa",
  "self": "http://localhost:8080/orders/5677459739508736",
  "size": "Medium"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
{
  "Error": "Order already exists."
}
Status: 401
{
  "Error": "Unauthorized. JWT not verified."
}
Status: 403
{
  "Error": "Forbidden. The request object has invalid data type."
}
Status: 404
{
  "Error": "Not Found. No drink with this drink_id exists."
}
Status: 406 Not Acceptable
{
  "Error": "Not Acceptable. MIME type is not supported."
}
```

List all Orders

List all the orders.

GET /orders

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns either HTML or JSON depending on accept header
Error	401 Unauthorized	JWT not verified
Error	415 Unsupported Media Type	MIME type is not supported

Response Examples

Success

Status: 200 OK (JSON)


```
{
    "orders": [
        {
            "date": "05-05-05",
            "order_id": 5149887163269120,
            "size": "Medium",
            "drink": 5081054809423872,
            "self": "http://localhost:8080/orders/5149887163269120",
            "owner": "auth0|63858def67ad3e09030cd3fa",
            "id": 5149887163269120
        },
        {
            "date": "05-05-05",
            "order_id": 5706627130851328,
            "size": "Medium",
            "drink": 5081054809423872,
            "self": "http://localhost:8080/orders/5706627130851328",
            "owner": "auth0|63858def67ad3e09030cd3fa",
            "id": 5706627130851328
        }
```

```
    ]
}
```

Failure

```
Status: 401
{
   "Error": "Unauthorized. JWT not verified."
}
Status: 415 Unsupported Media Type
{
   "Error": "Unsupported Media Type. Please submit 'application/json' or 'text/html' MIME type."
}
```

DELETE/PUT/PATCH Orders

Error.

[DELETE/PUT/PATCH] /orders

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	Response is not supported

Response Examples

Success

N/A

Failure

```
Status: 405 Not Found
{
   "Error": "Method Not Allowed. Allowed Methods: GET, POST."
}
```

GET a Order

Allows you to get an existing drink.

GET /orders/:order_id

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No order with this order_id exists
Failure	415 Unsupported Media Type	Media type is not supported

Response Examples

Success

```
Status: 200 OK (JSON)
{
   "date": "05-05-05",
   "order_id": 5702893864747008,
   "drink": 6243479118151680,
   "size": "Small",
   "self": "http://localhost:8080/orders/5702893864747008",
   "owner": "auth0|63858def67ad3e09030cd3fa"
}
```

Failure

```
Status: 404 Not Found
{
   "Error": "Not Found. No drink with this order_id exists."
}
Status: 415 Unsupported Media Type
{
   "Error":  " Unsupported Media Type. Please submit 'application/json' or 'text/html' MIME type."
}
```

Edit a Order

Allows you to edit a order.

PUT /orders/:order_id

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
date	Date	Yes
size	Size of the drink	Yes
drink_id	drink id	Yes

Request Body Example

```
{
   "date": "05-05-05",
   "size": "Small",
   "drink_id": 1234
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	401 Unauthorized	JWT not verified
Failure	403 Forbidden	Data type is incorrect.
Failure	404 Not Found	Drink ID does not exist.

Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 200 OK (JSON)

{
   "date": "05-05-05",
   "drink": 1234,
   "order_id": 5702893864747008,
   "owner": "auth0|63858def67ad3e09030cd3fa",
   "self": "http://localhost:8080/orders/5702893864747008",
   "size": "Small"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error":  "Bad Request. The request object is missing at least one of the required attributes"
}
{
   "Error": "Order already exists."
}
Status: 403
{
  "Error" : "Forbidden. The request object has invalid data type."
}

Status: 404 Not Found
{
   "Error": "Not Found. No drink with this drink_id exists."
}

Status: 406 Not Acceptable
{
  "Error": "Not Acceptable. MIME type is not supported."
}
```

Edit a Order

Allows you to edit a order.

PATCH /orders/:order_id

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
date	Date	No
size	Size of the drink	No
drink_id	drink id	No

Request Body Example

```
{
   "date": "05-05-05",
   "drink_id": 1234
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the drink must not be created, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	401 Unauthorized	JWT not verified
Failure	403 Forbidden	Data type is incorrect.
Failure	404 Not Found	Drink ID does not exist.

Failure	406 Not Acceptable	Invalid Accept header set

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

Success

```
Status: 200 OK (JSON)

{
   "date": "05-05-05",
   "drink": 1234,
   "order_id": 5702893864747008,
   "owner": "auth0|63858def67ad3e09030cd3fa",
   "self": "http://localhost:8080/orders/5702893864747008",
   "size": "Small"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error":  "Bad Request. The request object is missing at least one of the required attributes"
}
{
   "Error": "Order already exists."
}
Status: 403
{
  "Error" : "Forbidden. The request object has invalid data type."
}

Status: 404 Not Found
{
   "Error": "Not Found. No drink with this drink_id exists."
}

Status: 406 Not Acceptable
{
  "Error": "Not Acceptable. MIME type is not supported."
}
```

DELETE a Order

Allows you to delete an existing order. Corresponding drink is updated to show there is order_id is Null.

```
DELETE /orders/:order_id
```

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Body

None

Response

Response Body Format

JSON/HTML

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	404 Not Found	No order with this order_id exists

Response Examples

Success

```
Status: 204 OK (JSON)
```

Failure

```
Status: 404 Not Found
{
   "Error": "Not Found. No drink with this order_id exists."
}
```

Users

List all Users

List all the users.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns either HTML or JSON depending on accept header
Error	415 Unsupported Media Type	MIME type is not supported

Response Examples

Success

Status: 200 OK (JSON)


```
{
    "users": [
        {
            "user_id": 6524954094862336,
            "email": "wallace@cheese.com",
            "order": 6206230007644160,
            "user_sub": "auth0|63858def67ad3e09030cd3fa",
            "self": "http://localhost:8080//users/6524954094862336",
            "name": "wallace@cheese.com",
            "id": 6524954094862336
        }
    ]
}
```


Failure

Status: 415 Unsupported Media Type


```
{
   "Error": "Unsupported Media Type. Please submit 'application/json' or 'text/html' MIME type."
}
```