RESTful API Spec:

Cloud Application Development

Last Update: May 28. 11:15 am Pacific

Change log

Version	Change	Date
1.0	Initial version.	Jun 8, 2022

Data Model

The app stores three kinds of entities in Google Datastore, Users, Orders and Foods.

Users

Property	Data Type	Example Values	Notes
user_id	String	Generated by Datastore 325523745854	The unique ID of the user, which is the SUB property.
first_name	String	“John”	First name of the user.
Last_name	String	“Doe”	Last name of the user.
Comments: All properties are required. All properties are auto generated as well.			

NOTE: The user entity is automatically generated along with the properties when a user retrieves a new JWT token from the login page if the user entity doesn’t already exist in the datastore.

Orders (Protected)

Property	Data Type	Example Values	Notes
id	Integer	Generated by Datastore 325523745854	The id of the order. Datastore automatically generates it. Don't add it yourself as a property of the entity.
customer_id	String	Sub Property from JWT "11222222222222222"	The owner of the order. When creating a new order, automatically assigns the SUB property of the owner to it. Don’t add it yourself.
order_type	String	“To Go” “Dine In”	Type of order the order, e.g. Drive Through, Dine In, etc.
priority	String	“Urgent” “Now”	The urgency of getting the order completed. Fast, Urgent, Routine, etc.
amount	Integer	22.22, 51.41, 20, 100	How much the order costs.

items	Array	order[‘items’] = [{
		"calories": 1100,
		"name": "Chicken Enchiladas",
		"vegetarian": false,
		"id": 5116145900191744
		}
		]
			Contains the relationship(s) between the order and food items. Automatically generated.

Comments: All properties are required. id, customer_id, and items are auto generated on creating a new Order. It’s up to the user to fill in order_type, priority and amount fields in the request body.			

Foods

Property	Data Type	Example Values	Notes
id	Integer	Generated by Datastore 325523745854	The id of the order. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	“Pizza”, “Cob Salad”	The name of the food item.
calories	Integer	1200, 500, 250	The amount of Calories in the food (In Calories or kcal).
vegetarian	Boolean	true, false	If the food is prepared for vegetarians.

orders	Array	food[‘orders’] = [
		{
		"id": 5141618680135680
		}
		]
			Contains the relationship(s) between the food item and orders.
			Automatically generated.

Comments: All properties are required. id and orders are auto generated upon creating a new food item. It’s up to the user to fill in name, carloes and vegetarian fields in the request body.			

User and Non-User Entities:

- The User entity is the only User entity, while Orders and Foods are non-user entities.
- The User entity has a **unique identifier** in its ‘user_id’ property shown above in the table. This number is the SUB number correlated with an actual user when authenticated with a JWT.
- The Order entity is related to Users via its property of ‘customer_id’. This property is also a SUB number generated by user.
- When a user wants to create, read, update, delete with an Order entity, they must supply a JWT in the **response headers**. The JWT will be extracted when making a CRUD operation and will be checked for authentication and authorization. **User[‘user_id’] must match Order[‘customer_d’]** for a successful operation to take place.
- A user can be associated with many Orders.

Non-User Entities:

- Orders and Foods are related by two properties which hold information about the other entity object. Order[‘items’] contains all the foods that order is associated with. Food[‘orders’] contains all the Orders that Food Item is associated with.
- Orders can contain many Food entities. And Food Items can be associated with many Orders.
- If a user deletes one of their Orders, then all Food Items that were associated with that deleted Order, will no longer be associated with.
- If a user deletes a Food Item, then all Orders should not contain that Food anymore.

List all Users

Lists all the users in the datastore.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Response Examples

Success

Status: 200 OK

```
[
    {
        "user_id": "11222222222222222",
        "last_name": "Tyson",
        "first_name": "Mike"
    },
]
```

Failure

Status: 406 Not Acceptable

```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

Create a new Order (Protected)

Allows you to create a new Order.

POST /orders

Request

Path Parameters

None

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
order_type	Type of order the order, e.g. Drive Through, Dine In, etc.	Yes
priority	The urgency of getting the order completed. Fast, Urgent, Routine, etc.	Yes
amount	How much the order costs	Yes

Request Body Example

```
{
    "order_type": "Food Truck",
    "priority": "Quick",
    "amount": 22
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	406 Not Acceptable	Request accept header isn’t 'application/json' format.
Failure	415 Unsupported Media Type	Request body isn’t 'application/json' format.

Response Examples

Success

Status: 201 Created


```
{
    "priority": "Food Truck",
    "order_type": "Quick",
    "customer_id": "11222222222222222",
    "id": 4865928487501824,
    "items": [],
    "amount": 22,
    "self": "http://127.0.0.1:5000/orders/4865928487501824"
}
```


Failure

Status: 400 Bad Request


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


Status: 400 Bad Request


```
{
    "Error": "The request object contains an invalid attribute"
}
```


Status: 401 Unauthorized


```
{
    "Error": "Error: The JWT is invalid"
}
```


Status: 401 Unauthorized


```
{
    "Error": "No JWT provided"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```


Status: 415 Unsupported Media Type


```
{
    "Error": "The request body needs to be in JSON format"
}
```

Get an Order (Protected)

Allows the user to read the order if they generated the order.

GET /orders/:order_id

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	403 Forbidden	The order is owned by someone else.
Failure	404 Not Found	No order with the order_id exists.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Response Examples

Success

Status: 200 OK


```
{
    "priority": "later",
    "order_type": "door dash",
    "customer_id": "1122222222222222",
    "id": 4865928487501824,
    "items": [],
    "amount": 25,
    "self": "http://127.0.0.1:5000/orders/4865928487501824"
}
```


Failure

Status: 401 Unauthorized


```
{
```

<table>
<tr><td>

```
    "Error": "Error: The JWT is invalid"
}
```

</td></tr>
<tr><td>Status: 401 Unauthorized

```
{
    "Error": "No JWT provided"
}
```

</td></tr>
<tr><td>Status: 403 Forbidden

```
{
    "Error": "The order is owned by someone else"
}
```

</td></tr>
<tr><td>Status: 404 Not Found

```
{
    "Error": "No order with this order_id exists"
}
```

</td></tr>
<tr><td>Status: 406 Not Acceptable

```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

</td></tr>
</table>

List all Orders (Protected)

Lists all orders that belong to the user.

GET /orders

Request

Path Parameters

None

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Response Examples

Success

Status: 200 OK


```
{
    "orders": [
        {
            "priority": "Urgent",
            "customer_id": "11222222222222222",
            "order_type": "Dine In",
            "amount": 30,
            "id": 4725190999146496,
            "items": [],
            "self": "http://127.0.0.1:5000/orders/4725190999146496"
        },
        {
            "items": [],
            "order_type": "Drive Through",
            "customer_id": "11222222222222222",
            "priority": "Quick",
            "id": 5104575795167232,
```

```
            "amount": 15,
            "self": "http://127.0.0.1:5000/orders/5104575795167232"
        },
        {
            "priority": "Quick",
            "id": 5141618680135680,
            "items": [],
            "amount": 22,
            "customer_id": "11222222222222222",
            "order_type": "Food Truck",
            "self": "http://127.0.0.1:5000/orders/5141618680135680"
        },
        {
            "amount": 12,
            "id": 5712837116690432,
            "items": [],
            "priority": "Quick",
            "customer_id": "11222222222222222",
            "order_type": "Drive Through",
            "self": "http://127.0.0.1:5000/orders/5712837116690432"
        },
        {
            "amount": 120,
            "order_type": "Dine In",
            "priority": "Urgent",
            "items": [],
            "id": 5954557641228288,
            "customer_id": "11222222222222222",
            "self": "http://127.0.0.1:5000/orders/5954557641228288"
        }
    ],
    "total_items": 7,
    "next": "http://127.0.0.1:5000/orders?limit=5&offset=5"
}
```


Failure

Status: 401 Unauthorized


```
{
    "Error": "Error: The JWT is invalid"
}
```


Status: 401 Unauthorized


```
{
    "Error": "No JWT provided"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

Edit an Order (PATCH - Protected)

Allows the user to edit and order via patch if they generated the order.

PATCH /orders/:order_id

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
order_type	Type of order the order, e.g. Drive Through, Dine In, etc.	Optional
priority	The urgency of getting the order completed. Fast, Urgent, Routine, etc.	Optional
amount	How much the order costs	Optional

Request Body Example

```
{
    "order_type": "Uber Eats"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	403 Forbidden	The order is owned by someone else.
Failure	404 Not Found	No order with the order_id exists.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Failure	415 Unsupported Media Type	Request body isn't 'application/json' format.

Response Examples

Success

Status: 200 Ok


```
{
    "id": 4865928487501824,
    "amount": 150,
    "customer_id": "11222222222222222",
    "order_type": "Uber Eats",
    "priority": "Now",
    "items": [],
    "self": "http://127.0.0.1:5000/orders/4865928487501824"
}
```


Failure

Status: 400 Bad Request


```
{
    "Error": "The request object contains an invalid attribute"
}
```


Status: 401 Unauthorized


```
{
    "Error": "Error: The JWT is invalid"
}
```


Status: 401 Unauthorized


```
{
    "Error": "No JWT provided"
}
```


Status: 403 Forbidden


```
{
    "Error": "The order is owned by someone else"
}
```


Status: 404 Not Found


```
{
    "Error": "No order with this order_id exists"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```


Status: 415 Unsupported Media Type


```
{
    "Error": "The request body needs to be in JSON format"
}
```

Edit an Order (PUT - Protected)

Allows the user to edit and order via put if they generated the order.

PUT /orders/:order_id

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
order_type	Type of order the order, e.g. Drive Through, Dine In, etc.	Yes
priority	The urgency of getting the order completed. Fast, Urgent, Routine, etc.	Yes
amount	How much the order costs	Yes

Request Body Example

```
{
    "order_type": "Dine In",
    "priority": "Now",
    "amount": 150
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	403 Forbidden	The order is owned by someone else.

Failure	404 Not Found	No order with the order_id exists.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.
Failure	415 Unsupported Media Type	Request body isn't 'application/json' format.

Response Examples

Success

Status: 200 Ok


```
{
    "order_type": "Dine In",
    "id": 4865928487501824,
    "amount": 150,
    "priority": "Now",
    "customer_id": "11222222222222222",
    "items": [],
    "self": "http://127.0.0.1:5000/orders/4865928487501824"
}
```


Failure

Status: 400 Bad Request


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


Status: 400 Bad Request


```
{
    "Error": "The request object contains an invalid attribute"
}
```


Status: 401 Unauthorized


```
{
    "Error": "Error: The JWT is invalid"
}
```


Status: 401 Unauthorized


```
{
    "Error": "No JWT provided"
}
```


Status: 403 Forbidden


```
{
    "Error": "The order is owned by someone else"
}
```


Status: 404 Not Found


```
{
    "Error": "No order with this order_id exists"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```


Status: 415 Unsupported Media Type

```
{
    "Error": "The request body needs to be in JSON format"
}
```

Delete an Order (Protected)

Allows the user to delete the order if they generated it.

DELETE /orders/:order_id

Request

Path Parameters

Name	Description
order_id	ID of the order

Request Headers

Name	Description	Required?
Authorization	The end point must use an Authorization Header with the type of Bearer Token	Yes

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	No JWT is supplied or the JWT is invalid.
Failure	403 Forbidden	The order is owned by someone else.
Failure	404 Not Found	No order with the order_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized
{
"Error": "Error: The JWT is invalid"
}

Status: 401 Unauthorized
{
"Error": "No JWT provided"
}

Status: 403 Forbidden

```
{
    "Error": "The order is owned by someone else"
}
```

Status: 404 Not Found

```
{
    "Error": "No order with this order_id exists"
}
```

Create a new Food

Allows you to create a new food option.

POST /foods

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the food item.	Yes
calories	The amount of Calories in the food (In Calories or kcal).	Yes
vegetarian	If the food is prepared for vegetarians.	Yes

Request Body Example

```
{
    "name": "Polish Hotdog",
    "calories": 1200,
    "vegetarian": false
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.
Failure	415 Unsupported Media Type	Request body isn't 'application/json' format.

Response Examples

Success

Status: 201 Created

```
{
    "name": "Polish Hotdog",
    "calories": 1200,
    "vegetarian": false,
    "orders": [],
```

```
    "id": 4828657734385664,
    "self": "http://127.0.0.1:5000/foods/4828657734385664"
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
    "Error": "The request object contains an invalid attribute"
}
```

Status: 406 Not Acceptable

```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

Status: 415 Unsupported Media Type

```
{
    "Error": "The request body needs to be in JSON format"
}
```

Get a Food Item

Allows you to get an existing food item

GET /foods/:food_id

Request

Path Parameters

Name	Description
food_id	ID of the food item.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No food with this food_id exists
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Response Examples

Success

Status: 200 OK


```
{
    "name": "Polish Hotdog",
    "calories": 1200,
    "vegetarian": false,
    "orders": [],
    "id": 4828657734385664,
    "self": "http://127.0.0.1:5000/foods/4828657734385664"
}
```


Failure

Status: 404 Not Found


```
{
    "Error": "No food with this food_id exists"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

List all Foods

List all the foods.

GET /foods

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.

Response Examples

Success

Status: 200 OK


```
{
    "foods": [
        {
            "calories": 1200,
            "id": 4828657734385664,
            "name": "Polish Hotdog",
            "orders": [],
            "self": "http://127.0.0.1:5000/foods/4828657734385664",
            "vegetarian": false
        },
        {
            "calories": 1500,
            "id": 4862202200719360,
            "name": "Vanilla Milk Shake",
            "orders": [],
            "self": "http://127.0.0.1:5000/foods/4862202200719360",
            "vegetarian": true
        },
        {
            "calories": 190,
            "id": 4894284834668544,
            "name": "Fruit Salad",
            "orders": [],
            "self": "http://127.0.0.1:5000/foods/4894284834668544",
```

```
            "vegetarian": true
        },
        {
            "calories": 400,
            "id": 5139943911325696,
            "name": "French Fries",
            "orders": [],
            "self": "http://127.0.0.1:5000/foods/5139943911325696",
            "vegetarian": true
        },
        {
            "calories": 600,
            "id": 6267518586978304,
            "name": "Grilled Cheese Sandwich",
            "orders": [],
            "self": "http://127.0.0.1:5000/foods/6267518586978304",
            "vegetarian": true
        }
    ],
    "next": "http://127.0.0.1:5000/foods?limit=5&offset=5",
    "total_items": 6
}
```


Failure

Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```

Edit a Food Item (PATCH)

Allows the user to edit a food item.

PATCH /foods/:food_id

Request

Path Parameters

Name	Description
food_id	ID of the food item

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the food item.	Optional
calories	The amount of Calories in the food (In Calories or kcal).	Optional
vegetarian	If the food is prepared for vegetarians.	Optional

Request Body Example

```
{
    "vegetarian": "false"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	404 Not Found	No food with this food_id exists.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.
Failure	415 Unsupported Media Type	Request body isn't 'application/json' format.

Response Examples

Success

Status: 200 Ok

```
{
    "id": 6267518586978304,
    "orders": [],
    "vegetarian": false,
```

```
    "name": "Grilled Cheese Sandwich",
    "calories": 600,
    "self": "http://127.0.0.1:5000/foods/6267518586978304"
}
```


Failure

Status: 400 Bad Request


```
{
    "Error": "The request object contains an invalid attribute"
}
```


Status: 404 Not Found


```
{
    "Error": "No food with this food_id exists"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```


Status: 415 Unsupported Media Type


```
{
    "Error": "The request body needs to be in JSON format"
}
```

Edit a Food Item (PUT)

Allows the user to edit a food item.

PATCH /foods/:food_id

Request

Path Parameters

Name	Description
food_id	ID of the food item

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the food item.	Yes
calories	The amount of Calories in the food (In Calories or kcal).	Yes
vegetarian	If the food is prepared for vegetarians.	Yes

Request Body Example

```
{
    "name": "Chocolate Cake",
    "calories": 560,
    "vegetarian": true
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing one of the required attributes or one of the attributes is invalid.
Failure	404 Not Found	No food with this food_id exists.
Failure	406 Not Acceptable	Request accept header isn't 'application/json' format.
Failure	415 Unsupported Media Type	Request body isn't 'application/json' format.

Response Examples

Success

Status: 200 Ok

```
{
    "id": 6267518586978304,
```

```
    "orders": [],
    "calories": 560,
    "vegetarian": true,
    "name": "Chocolate Cake",
    "self": "http://127.0.0.1:5000/foods/6267518586978304"
}
```


Failure

Status: 400 Bad Request


```
{
    "Error": "The request object contains an invalid attribute"
}
```


Status: 400 Bad Request


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


Status: 404 Not Found


```
{
    "Error": "No food with this food_id exists"
}
```


Status: 406 Not Acceptable


```
{
    "Error": "Server is unable to provide the Content-Type requested"
}
```


Status: 415 Unsupported Media Type


```
{
    "Error": "The request body needs to be in JSON format"
}
```

Delete a Food Item

Allows you to delete a food item.

DELETE /foods/:food_id

Request

Path Parameters

Name	Description
food_id	ID of the food_item

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No food with this food_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
    "Error": "No food with this food_id exists"
}
```

Add a Food Item to an Order

Creates a relationship between an order and food item.

PATCH /orders/:order_id/foods/:food_id

Request

Path Parameters

Name	Description
order_id	ID of the order
food_id	ID of the food item

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Successfully added a relationship between an Order and Food Item.
Failure	403 Forbidden	Food Item already is on the order.
Failure	404 Not Found	The Order with the order_id or the Food with the food_id doesn't exist.

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

```
{
    "Error": "That food is already on the order"
}
```

Status: 404 Not Found

```
{
    "Error": "No order with this order_id exists"
}
```

Status: 404 Not Found

```
{
    "Error": "No food with this food_id exists"
}
```

Remove a Food Item from an Order

Boat has left the slip to go to sea. The slip is now empty.

DELETE /orders/:order_id/foods/:food_id

Request

Path Parameters

Name	Description
order_id	ID of the order
food_id	ID of the food item

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Removes the relationship between an Order and Food Item.
Failure	404 Not Found	The Order with the order_id or the Food with the food_id doesn't exist.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
    "Error": "No order with this order_id exists"
}
```

Status: 404 Not Found

```
{
    "Error": "No food with this food_id exists"
}
```