Gather Green Server API Methods:
Postman, postico, and postgres are required to test this server API locally.

Open your terminal, go to GatherGreenServer Directory, then run `vapor run migrate`.
If you meet the error saying ‘ `error: no such module 'PackageDescription'`
Try this command, `sudo xcode-select -s /Applications/Xcode.app.`

* Hopefully, the server can be deployed to VCM before the end of this week (later you can change the beginning part of the hyperlink to http://sw572.colab.duke.edu .
* A toy ios app for testing this server will be provided soon.

* The HTTP request type is indicated by http://127.0.0.1:8080/tablename/requesttype. Don’t forget to set the correct type when you test the API in postman.

* If you have any requirements, please let me know. I can change the API output accordingly.

PRODUCT
1. Post New Product
http://127.0.0.1:8080/products/post
Header:
Content-Type: application/json
Body:
```
{
  "name": "Newly collected Paper",
  "type": "Paper",
  "statue": "onsale",
  "permission": "standard",
  "description": "This is.a piece of paper!",
  "price": 10.00,
  "image": "paper",
  "volume": 10
}
```
Return: (json file with id generated by the database)
```
{
  "statue": "onsale",
  "volume": 10,
  "permission": "standard",
  "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
  "price": 10,
  "image": "paper",
  "type": "Paper",
  "description": "This is.a piece of paper!",
  "name": "Newly collected Paper"
}
```

2. Update existing Product
* id is required for the json file. Use “get” to get the id first before updating the product.
* here, the id is the one generated by my database, if you are testing, please use postico to substitute the id in sample body with the actual one

http://127.0.0.1:8080/products/put
Header:
Content-Type: application/json
Body: (this example changes the price of the newly collected cardboard from 10 to 25.)

```
{
    "statue": "onsale",
    "volume": 10,
    "permission": "standard",
    "id": "bd95b3a5-a971-4ea1-b0e0-297f2b4297d8",
    "price": 25,
    "image": "cardboard",
    "description": "This is.a cardboard!",
    "type": "CardBoard",
    "name": "Newly collected CardBoard"
  }
```

Return:
If success: 200 OK

3. Get Existing Product

http://127.0.0.1:8080/products/get

Return: (I have three products in my database while I am testing)

```
[
  {
    "statue": "onsale",
    "volume": 10,
    "permission": "standard",
    "id": "7D86E91F-D558-4CFE-8496-B330571AD4B0",
    "price": 10,
    "image": "paper",
    "description": "This is.a piece of paper!",
    "type": "Paper",
    "name": "Newly collected Paper"
  },
  {
    "statue": "onsale",
    "volume": 10,
    "permission": "standard",
    "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
    "price": 10,
    "image": "paper",
    "description": "This is.a piece of paper!",
    "type": "Paper",
    "name": "Newly collected Paper"
  },
  {
    "statue": "onsale",
    "volume": 10,
    "permission": "standard",
    "id": "BD95B3A5-A971-4EA1-B0E0-297F2B4297D8",
    "price": 25,
    "image": "cardboard",
    "description": "This is.a cardboard!",
    "type": "CardBoard",
    "name": "Newly collected CardBoard"
  }
]
```

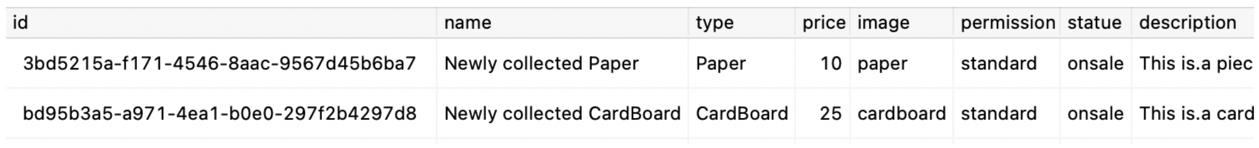
4. Delete Existing Product

http://127.0.0.1:8080/products/productid

Here, I use http://127.0.0.1:8080/products/7d86e91f-d558-4cfe-8496-b330571ad4b0

Return:
If success: 200 OK

Now, my product table looks like this (only two products available):



id	name	type	price	image	permission	statue	description
3bd5215a-f171-4546-8aac-9567d45b6ba7	Newly collected Paper	Paper	10	paper	standard	onsale	This is.a piec
bd95b3a5-a971-4ea1-b0e0-297f2b4297d8	Newly collected CardBoard	CardBoard	25	cardboard	standard	onsale	This is.a card

USER

As for now, the user table is just for demo purposes, no authentication functionality available (no hash).

1. Create New User
http://127.0.0.1:8080/users/post
Header:
Content-Type: application/json
Body:

```
{
  "role":"standard",
  "firstname": "John",
  "lastname": "Doe",
  "username": "IamJD",
  "email":"jd101@duke.edu",
  "password": "123456",
  "address": "Durham NC, 27710",
  "payment":"bankaccount1012"
}
```

Return (with id):

```
{
  "address": "Durham NC, 27710",
  "password": "123456",
  "id": "898FA9A7-55FA-4743-BD51-4254C9451F42",
  "lastname": "Doe",
  "payment": "bankaccount1012",
  "role": "standard",
  "email": "jd101@duke.edu",
  "firstname": "John",
  "username": "IamJD"
}
```

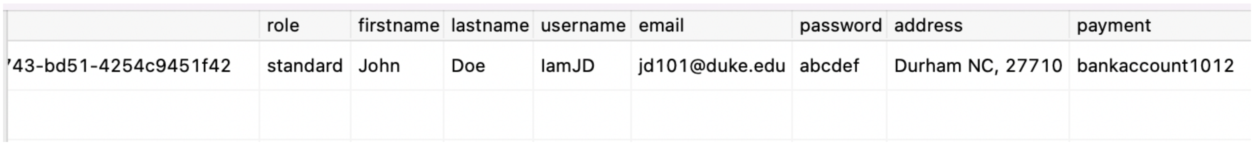
2. Update Existing User
* id is required.

http://127.0.0.1:8080/users/put
Header:
Content-Type: application/json
Body: (this example changes the password from 123456 to abcdef.)

```
{
  "role":"standard",
  "firstname": "John",
  "id": "898FA9A7-55FA-4743-BD51-4254C9451F42",
  "lastname": "Doe",
  "username": "IamJD",
  "email":"jd101@duke.edu",
  "password": "abcdef",
  "address": "Durham NC, 27710",
  "payment":"bankaccount1012"
}
```

Return:
If success: 200 OK

Now, my user table looks like this:



	role	firstname	lastname	username	email	password	address	payment
'43-bd51-4254c9451f42	standard	John	Doe	IamJD	jd101@duke.edu	abcdef	Durham NC, 27710	bankaccount1012

3. Get Existing User
http://127.0.0.1:8080/users/get
Return:

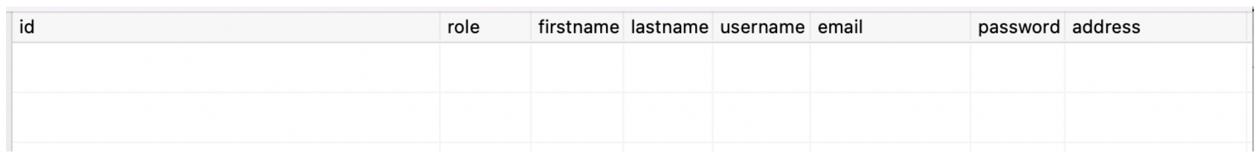
```
[
  {
    "address": "Durham NC, 27710",
    "password": "123456",
    "id": "898FA9A7-55FA-4743-BD51-4254C9451F42",
    "lastname": "Doe",
    "role": "standard",
    "firstname": "John",
    "email": "jd101@duke.edu",
    "username": "IamJD",
    "payment": "bankaccount1012"
  }
]
```

4. Delete Existing User

http://127.0.0.1:8080/users/userid

Here, I use http://127.0.0.1:8080/products/898fa9a7-55fa-4743-bd51-4254c9451f42
Return:
If success: 200 OK

Now, my user table is empty:



id	role	firstname	lastname	username	email	password	address

INVENTORY

1. Add to Inventory (after admin's approval)

http://127.0.0.1:8080/inventories/post

Header:

Content-Type: application/json

Body: (the product is get by GET method to product table)

```
{
  "product": {
     "statue": "onsale",
     "volume": 10,
     "permission": "standard",
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
     "price": 10,
     "image": "paper",
     "type": "Paper",
     "description": "This is.a piece of paper!",
     "name": "Newly collected Paper"
  },
  "quantity": 2.5    //lbs
}
```

Return: you will get the id for this inventory

```
{
  "product": {
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7"
  },
  "quantity": 2.5,
  "id": "B3DE5A0D-D18F-4694-8C63-BA26A169EF3B"
}
```

2. Update Existing Inventory

* id is required.
* One can only change the quantity.

http://127.0.0.1:8080/inventories/put

Header:

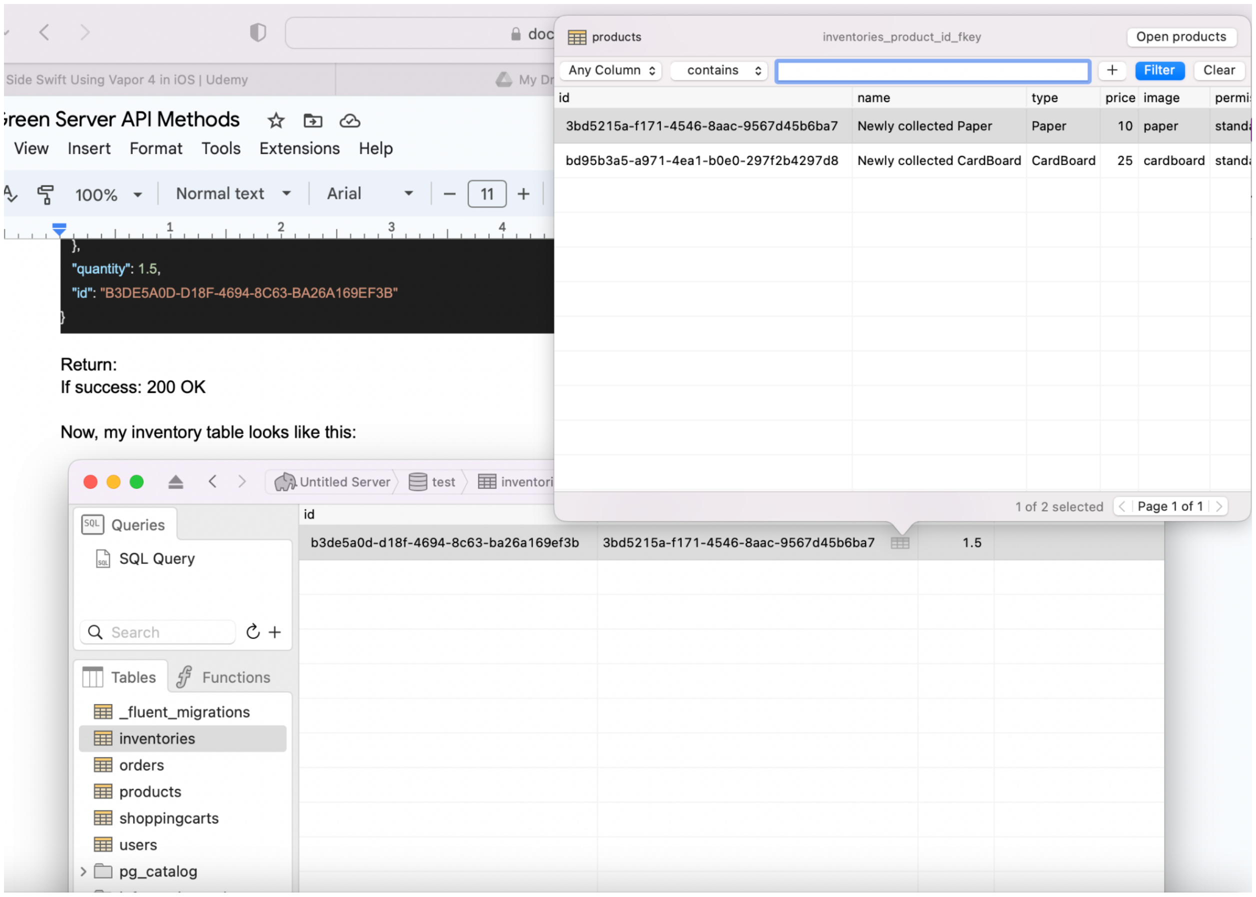
Content-Type: application/json

Body: (this example changes the quantity of this product from 2.5 lbs to 1.5 lbs.)

```
{
  "product": {
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7"
  },
  "quantity": 1.5,
  "id": "B3DE5A0D-D18F-4694-8C63-BA26A169EF3B"
}
```

Return:
If success: 200 OK

Now, my inventory table looks like this:
You can see the foreign key is working and the quantity has been changed to 1.5 lbs.



```
},
"quantity": 1.5,
"id": "B3DE5A0D-D18F-4694-8C63-BA26A169EF3B"
}
```

Return:
If success: 200 OK

Now, my inventory table looks like this:

products — inventories_product_id_fkey

id	name	type	price	image	permi
3bd5215a-f171-4546-8aac-9567d45b6ba7	Newly collected Paper	Paper	10	paper	stand
bd95b3a5-a971-4ea1-b0e0-297f2b4297d8	Newly collected CardBoard	CardBoard	25	cardboard	stand

1 of 2 selected — Page 1 of 1

id		
b3de5a0d-d18f-4694-8c63-ba26a169ef3b	3bd5215a-f171-4546-8aac-9567d45b6ba7	1.5

Tables: _fluent_migrations, inventories, orders, products, shoppingcarts, users, pg_catalog

3. Get Existing Inventory

http://127.0.0.1:8080/inventories/get

Return:

I joined two tables together.
```
[
  {
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "description": "This is.a piece of paper!",
      "type": "Paper",
      "name": "Newly collected Paper"
    },
    "quantity": 2.5,
    "id": "5B2E1873-D5DF-477D-99A7-3C45478642A0"
  }
]
```

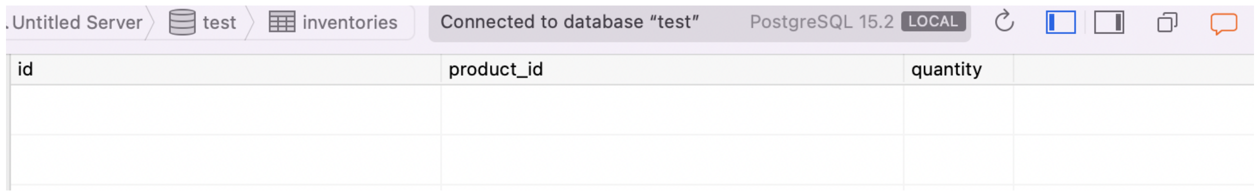
4. Delete Existing Inventory

http://127.0.0.1:8080/inventories/inventoryid

Here, I use http://127.0.0.1:8080/inventories/b3de5a0d-d18f-4694-8c63-ba26a169ef3b

Return:
If success: 200 OK

Now, my inventories table is empty:



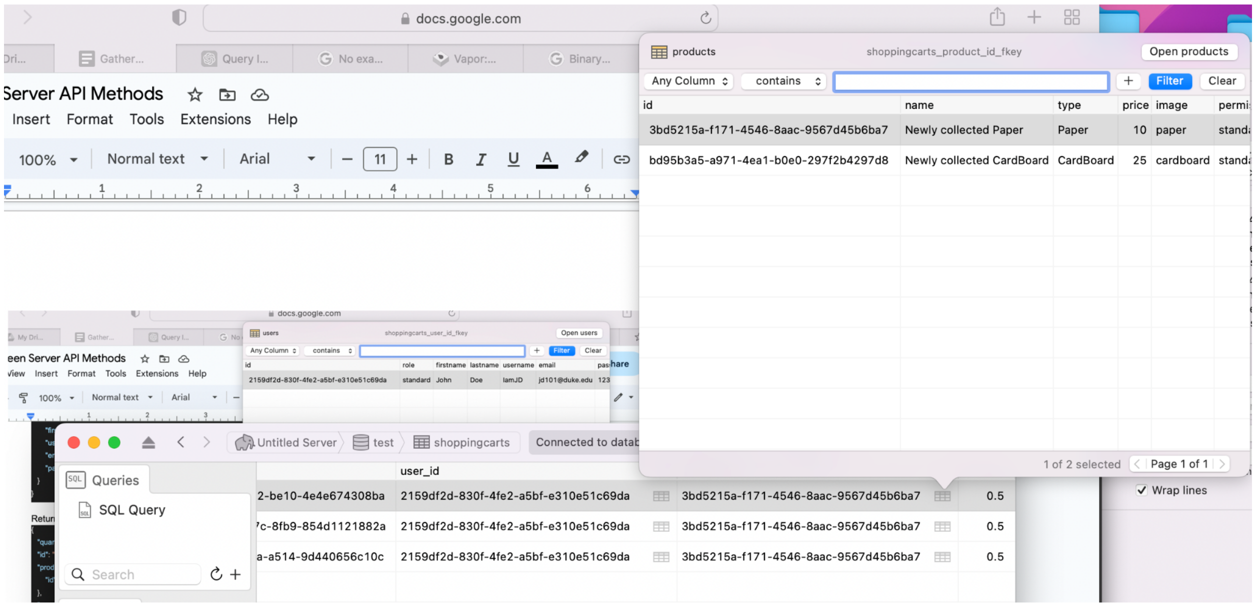
id	product_id	quantity

SHOPPING CART
1. Add A New Product to Shopping Cart
http://127.0.0.1:8080/shoppingcarts/post
Header:
Content-Type: application/json
Body:
```
{
  "product": {
    "statue": "onsale",
    "volume": 10,
    "permission": "standard",
    "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
    "price": 10,
    "image": "paper",
    "type": "Paper",
    "description": "This is.a piece of paper!",
    "name": "Newly collected Paper"
  },
  "quantity": 0.5,
  "user": {
    "address": "Durham NC, 27710",
    "password": "123456",
    "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
    "lastname": "Doe",
    "role": "standard",
    "firstname": "John",
    "username": "IamJD",
    "email": "jd101@duke.edu",
    "payment": "bankaccount1012"
  }
}
```

Return: (with a newly generated id)

```
{
  "quantity": 0.5,
  "id": "24F86D52-874B-43E2-BE10-4E4E674308BA",
  "product": {
    "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7"
  },
  "user": {
    "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
  }
}
```

In the shopping cart table, linking is working perfectly.
Link to products table:



products — shoppingcarts_product_id_fkey

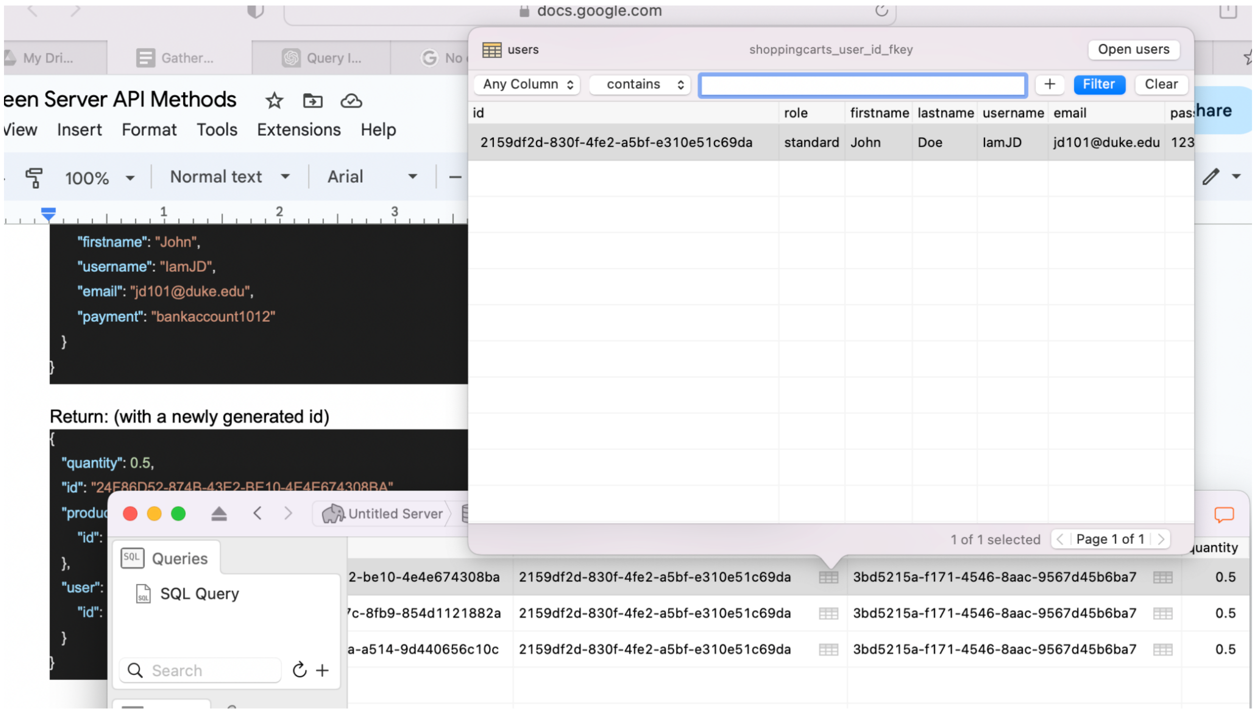
id	name	type	price	image	permi
3bd5215a-f171-4546-8aac-9567d45b6ba7	Newly collected Paper	Paper	10	paper	standa
bd95b3a5-a971-4ea1-b0e0-297f2b4297d8	Newly collected CardBoard	CardBoard	25	cardboard	standa

1 of 2 selected Page 1 of 1

Untitled Server › test › shoppingcarts Connected to datab

	user_id		
2-be10-4e4e674308ba	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
7c-8fb9-854d1121882a	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
a-a514-9d440656c10c	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5

Link to user table:



users — shoppingcarts_user_id_fkey

id	role	firstname	lastname	username	email	pas
2159df2d-830f-4fe2-a5bf-e310e51c69da	standard	John	Doe	lamJD	jd101@duke.edu	123

1 of 1 selected Page 1 of 1

```
    "firstname": "John",
    "username": "lamJD",
    "email": "jd101@duke.edu",
    "payment": "bankaccount1012"
  }
}
```

Return: (with a newly generated id)

```
{
  "quantity": 0.5,
  "id": "24F86D52-874B-43F2-BE10-4E4E674308BA",
  "produ
    "id":
  },
  "user":
    "id":
  }
}
```

			quantity
2-be10-4e4e674308ba	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
7c-8fb9-854d1121882a	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
a-a514-9d440656c10c	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5

2. Update Existing Shopping Cart Item

* id is required.

* One can only change the quantity.

http://127.0.0.1:8080/shoppingcarts/put

Header:

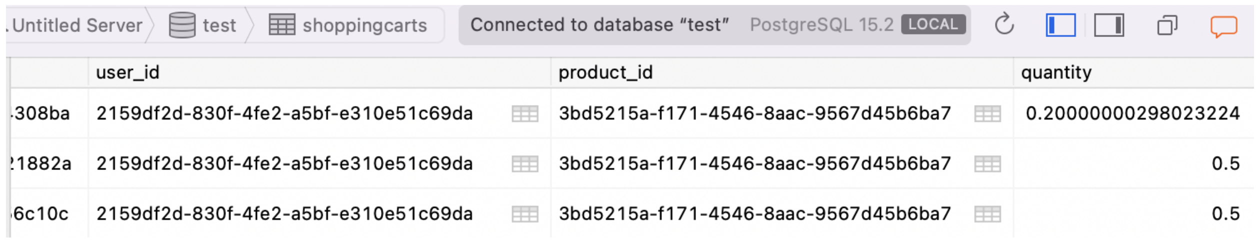
Content-Type: application/json

Body: (this example changes the quantity of this product from 0.5 lbs to 0.2 lbs.)

```
{
  "quantity": 0.2,
  "id": "24F86D52-874B-43E2-BE10-4E4E674308BA",
  "product": {
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7"
  },
  "user": {
     "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
  }
}
```

Return:
If success: 200 OK

Now, my shopping cart table is:



.Untitled Server 〉 test 〉 shoppingcarts   Connected to database "test"   PostgreSQL 15.2 LOCAL

	user_id	product_id	quantity
308ba	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.20000000298023224
1882a	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
6c10c	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5

3. Get Existing Shopping Cart for a Specific User

http://127.0.0.1:8080/shoppingcarts/get
Header:
Content-Type: application/json
Body:

```
{
     "address": "Durham NC, 27710",
     "password": "123456",
     "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
     "lastname": "Doe",
     "role": "standard",
     "firstname": "John",
     "username": "IamJD",
     "email": "jd101@duke.edu",
     "payment": "bankaccount1012"
  }
```

Return:

I joined two (shopping cart and product) tables together. And I have three items in my shopping cart for John Doe.

```
[
  {
    "quantity": 0.20000000298023224,
    "id": "24F86D52-874B-43E2-BE10-4E4E674308BA",
    "product": {
      "volume": 10,
      "statue": "onsale",
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "description": "This is.a piece of paper!",
      "type": "Paper",
      "name": "Newly collected Paper"
    },
    "user": {
      "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
    }
  },
  {
    "quantity": 0.5,
    "id": "7E21317F-899E-468A-A514-9D440656C10C",
    "product": {
      "volume": 10,
      "statue": "onsale",
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "description": "This is.a piece of paper!",
      "type": "Paper",
      "name": "Newly collected Paper"
    },
    "user": {
      "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
    }
  },
  {
    "quantity": 0.5,
    "id": "659165A0-4760-4B7C-8FB9-854D1121882A",
    "product": {
      "volume": 10,
      "statue": "onsale",
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "description": "This is.a piece of paper!",
      "type": "Paper",
      "name": "Newly collected Paper"
    },
    "user": {
      "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
    }
  }
]
```

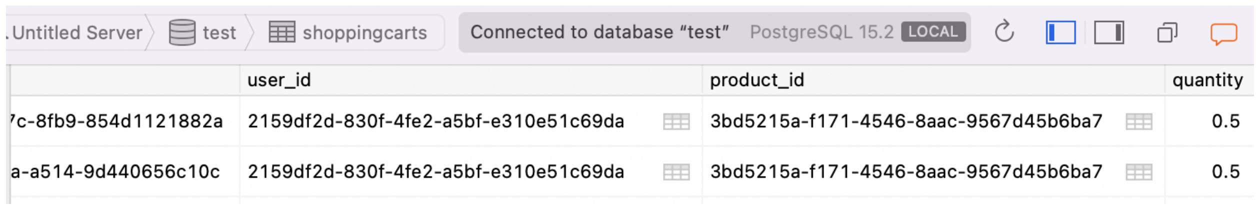
4. Delete An Existing Shopping Cart Item

http://127.0.0.1:8080/shoppingcarts/shoppingcartid

Here, I use http://127.0.0.1:8080/shoppingcarts/24F86D52-874B-43E2-BE10-4E4E674308BA
I used all uppercase version of the id, it still works perfectly. It means the id part of the link is case-insensitive. One can just use the id returned by the GET method straightly.
The shopping cart item with the quantity 0.20.

Return:
If success: 200 OK

Now, my shopping cart table is:



	user_id	product_id	quantity
7c-8fb9-854d1121882a	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5
a-a514-9d440656c10c	2159df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5

ORDER:
For the order table, you can only post, get one, get all. Any other change to the order table is invalid. In our system, once you've placed an order. You can no longer change it.

To test, I first created another user Bobby Brown with randomly generated ID "0C9B7CEA-6787-430B-952A-FEF394B73871".

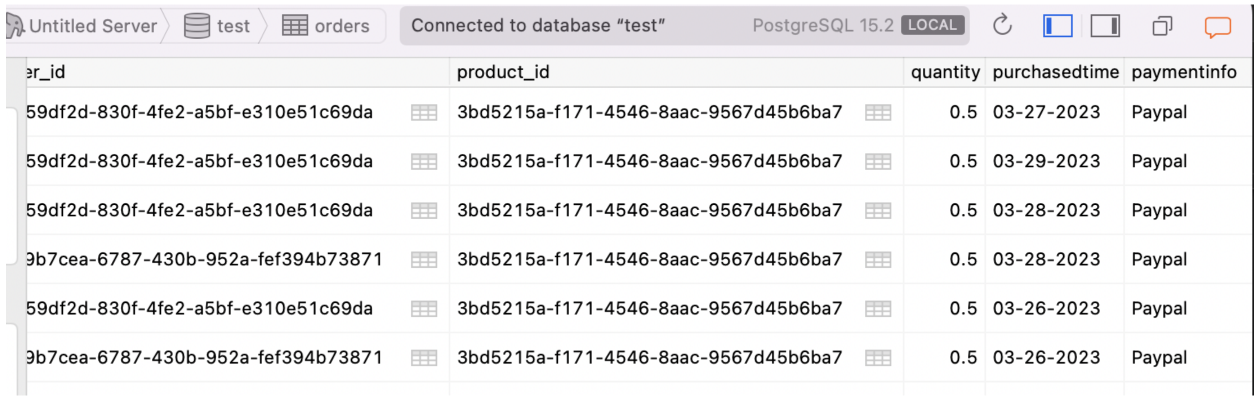
1. Add A New Order to Orders Table
http://127.0.0.1:8080/shoppingcarts/post
Header:
Content-Type: application/json
Body:

```
{
  "product": {
     "statue": "onsale",
     "volume": 10,
     "permission": "standard",
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
     "price": 10,
     "image": "paper",
     "type": "Paper",
     "description": "This is.a piece of paper!",
     "name": "Newly collected Paper"
  },
  "quantity": 0.5,
  "user": {
     "address": "Durham NC, 27710",
     "password": "123456",
     "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
     "lastname": "Doe",
     "role": "standard",
     "firstname": "John",
     "username": "IamJD",
     "email": "jd101@duke.edu",
     "payment": "bankaccount1012"
  },
  "purchasedtime": "03-29-2023",
  "paymentinfo": "Paypal"
}
```

Return: (with an order id)

```
{
  "paymentinfo": "Paypal",
  "quantity": 0.5,
  "id": "3E11D289-D6CA-4E91-8D2C-57DE6D269D3F",
  "purchasedtime": "03-29-2023",
  "product": {
     "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7"
  },
  "user": {
     "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA"
  }
}
```

After posting 5 orders for two different customers (John Doe and Bobby Brown):



Untitled Server › test › orders   Connected to database "test"   PostgreSQL 15.2 LOCAL

er_id	product_id	quantity	purchasedtime	paymentinfo
59df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-27-2023	Paypal
59df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-29-2023	Paypal
59df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-28-2023	Paypal
9b7cea-6787-430b-952a-fef394b73871	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-28-2023	Paypal
59df2d-830f-4fe2-a5bf-e310e51c69da	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-26-2023	Paypal
9b7cea-6787-430b-952a-fef394b73871	3bd5215a-f171-4546-8aac-9567d45b6ba7	0.5	03-26-2023	Paypal

2. Get All Orders for All Customers

http://127.0.0.1:8080/orders/getall

Return:

```
[
  {
     "quantity": 0.5,
     "id": "3E11D289-D6CA-4E91-8D2C-57DE6D269D3F",
     "paymentinfo": "Paypal",
     "product": {
        "statue": "onsale",
        "volume": 10,
        "permission": "standard",
        "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
        "price": 10,
        "image": "paper",
        "type": "Paper",
        "description": "This is.a piece of paper!",
        "name": "Newly collected Paper"
     },
     "purchasedtime": "03-29-2023",
     "user": {
        "address": "Durham NC, 27710",
        "password": "123456",
        "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
        "payment": "bankaccount1012",
        "lastname": "Doe",
        "role": "standard",
        "username": "IamJD",
        "email": "jd101@duke.edu",
        "firstname": "John"
    }
},
{
    "quantity": 0.5,
    "id": "6FB60DB9-DD5A-433A-96A0-F45051D15BCA",
    "paymentinfo": "Paypal",
    "product": {
        "statue": "onsale",
        "volume": 10,
        "permission": "standard",
        "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
        "price": 10,
        "image": "paper",
        "type": "Paper",
        "description": "This is.a piece of paper!",
        "name": "Newly collected Paper"
    },
    "purchasedtime": "03-28-2023",
    "user": {
        "address": "Durham NC, 27710",
        "password": "123456",
        "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
        "payment": "bankaccount1012",
        "lastname": "Doe",
        "role": "standard",
        "username": "IamJD",
        "email": "jd101@duke.edu",
        "firstname": "John"
    }
},
{
    "quantity": 0.5,
    "id": "2B1897DD-552A-47D1-ADDE-564B605982A7",
    "paymentinfo": "Paypal",
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "type": "Paper",
      "description": "This is.a piece of paper!",
      "name": "Newly collected Paper"
    },
    "purchasedtime": "03-27-2023",
    "user": {
      "address": "Durham NC, 27710",
      "password": "123456",
      "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
      "payment": "bankaccount1012",
      "lastname": "Doe",
      "role": "standard",
      "username": "IamJD",
      "email": "jd101@duke.edu",
      "firstname": "John"
    }
  },
  {
    "quantity": 0.5,
    "id": "960A79BB-4F1F-4A87-B3F5-CCD5FF87433F",
    "paymentinfo": "Paypal",
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "type": "Paper",
      "description": "This is.a piece of paper!",
      "name": "Newly collected Paper"
    },
    "purchasedtime": "03-26-2023",
    "user": {
      "address": "Durham NC, 27710",
            "password": "123456",
            "id": "2159DF2D-830F-4FE2-A5BF-E310E51C69DA",
            "payment": "bankaccount1012",
            "lastname": "Doe",
            "role": "standard",
            "username": "IamJD",
            "email": "jd101@duke.edu",
            "firstname": "John"
        }
    },
    {
        "quantity": 0.5,
        "id": "D650CE6F-00D0-4E5D-9693-B64B79B9313D",
        "paymentinfo": "Paypal",
        "product": {
            "statue": "onsale",
            "volume": 10,
            "permission": "standard",
            "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
            "price": 10,
            "image": "paper",
            "type": "Paper",
            "description": "This is.a piece of paper!",
            "name": "Newly collected Paper"
        },
        "purchasedtime": "03-26-2023",
        "user": {
            "address": "Durham NC, 27710",
            "password": "123456",
            "id": "0C9B7CEA-6787-430B-952A-FEF394B73871",
            "payment": "bankaccount1012",
            "lastname": "Brown",
            "role": "standard",
            "username": "IamBobby",
            "email": "bb101@duke.edu",
            "firstname": "Bobby"
        }
    },
    {
        "quantity": 0.5,
        "id": "7DD4848B-CB6B-4A1D-B84D-35F485325BB3",
        "paymentinfo": "Paypal",
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "type": "Paper",
      "description": "This is.a piece of paper!",
      "name": "Newly collected Paper"
    },
    "purchasedtime": "03-28-2023",
    "user": {
      "address": "Durham NC, 27710",
      "password": "123456",
      "id": "0C9B7CEA-6787-430B-952A-FEF394B73871",
      "payment": "bankaccount1012",
      "lastname": "Brown",
      "role": "standard",
      "username": "IamBobby",
      "email": "bb101@duke.edu",
      "firstname": "Bobby"
    }
  }
]
```

3. Get All Orders for A Specific Customer

http://127.0.0.1:8080/orders/getone

Header:
Content-Type: application/json
Body:

```
{
  "address": "Durham NC, 27710",
  "password": "123456",
  "id": "0C9B7CEA-6787-430B-952A-FEF394B73871",
  "lastname": "Brown",
  "payment": "bankaccount1012",
  "role": "standard",
  "username": "IamBobby",
  "firstname": "Bobby",
  "email": "bb101@duke.edu"
}
```

Return: (Two orders that Bobby Brown placed)

```
[
  {
    "quantity": 0.5,
    "id": "7DD4848B-CB6B-4A1D-B84D-35F485325BB3",
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "description": "This is.a piece of paper!",
      "type": "Paper",
      "name": "Newly collected Paper"
    },
    "purchasedtime": "03-28-2023",
    "paymentinfo": "Paypal",
    "user": {
      "id": "0C9B7CEA-6787-430B-952A-FEF394B73871"
    }
  },
  {
    "paymentinfo": "Paypal",
    "quantity": 0.5,
    "id": "D650CE6F-00D0-4E5D-9693-B64B79B9313D",
    "purchasedtime": "03-26-2023",
    "product": {
      "statue": "onsale",
      "volume": 10,
      "permission": "standard",
      "id": "3BD5215A-F171-4546-8AAC-9567D45B6BA7",
      "price": 10,
      "image": "paper",
      "type": "Paper",
      "description": "This is.a piece of paper!",
      "name": "Newly collected Paper"
    },
      "user": {
         "id": "0C9B7CEA-6787-430B-952A-FEF394B73871"
      }
   }
]
```