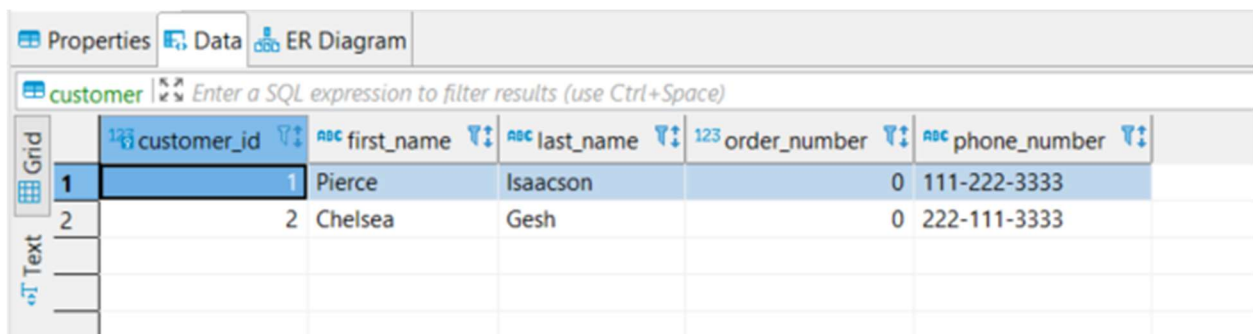


<https://github.com/Piercelsaacson/week-18-Spring-week6FinalProject>

My project is only functional to performing all crud operations to one table through the API. While in the final block of the course, the API we were learning along with in the vidoes gave me feedback the asignments required but when attempting more detailed requests, something was fundamentally wrong with my code that I could not nail down. I tried creating a new project with the help of online tutorials but I could not get the point of the project working and only made a connection to one of my five tables.

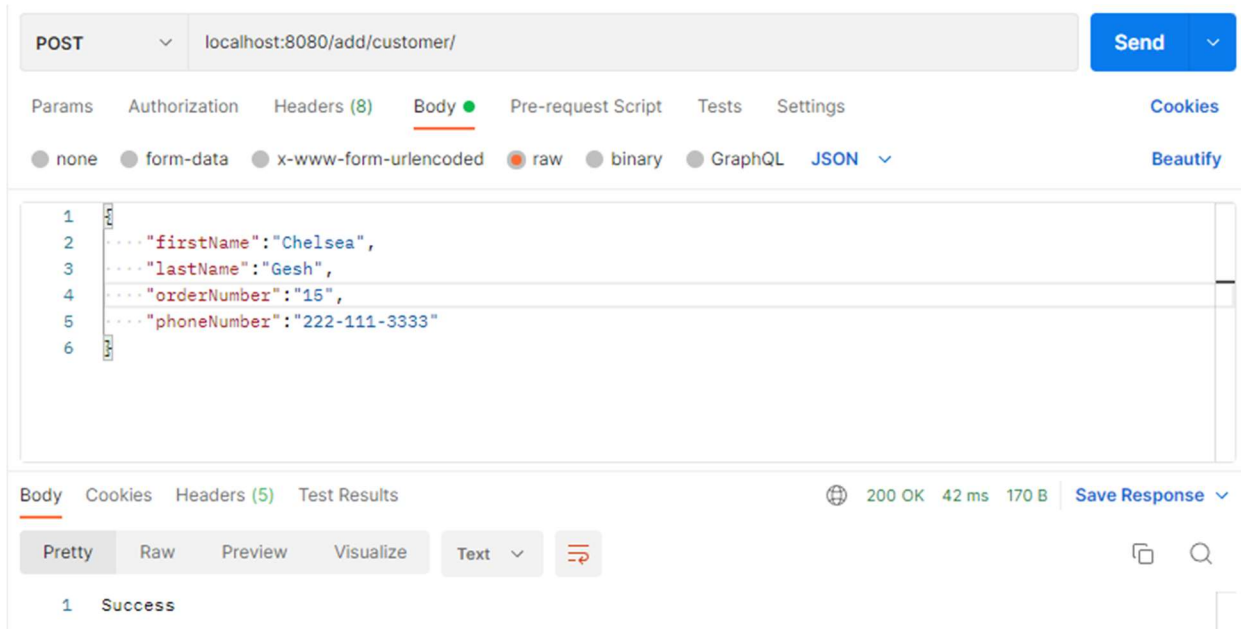
Initial view of customer table with data



customer				
Enter a SQL expression to filter results (use Ctrl+Space)				
customer_id	first_name	last_name	order_number	phone_number
1	Pierce	Isaacson	0	111-222-3333
2	Chelsea	Gesh	0	222-111-3333

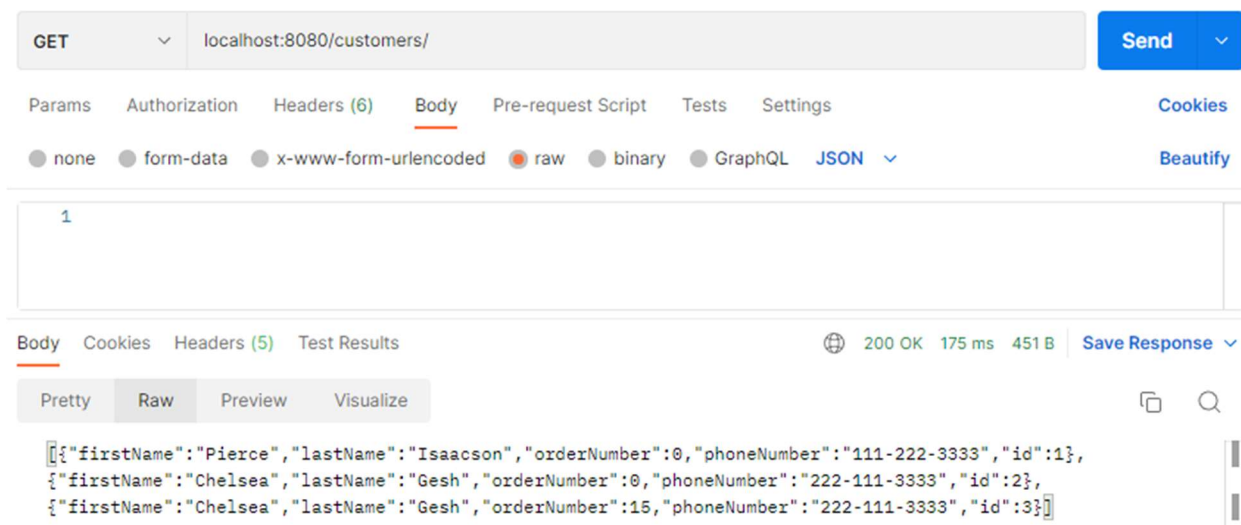
Using Postman

URL post request adding a new customer (redundent info to be updated later in example)



POST/add request successful

GET/Retrieve all request of what is in the table



GET/Retrieve successful

GET/Retrieval of one customer

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8080/customer/1/
- Send Button:** A blue button labeled "Send" with a dropdown arrow.
- Tabs:** Params, Authorization, Headers (6), **Body** (selected), Pre-request Script, Tests, Settings.
- Body Type:** none, form-data, x-www-form-urlencoded, **raw** (selected), binary, GraphQL, JSON (dropdown).
- Body Content:** A large empty text area with the number "1" at the top left.
- Status Bar:** 200 OK, 24 ms, 260 B, Save Response (dropdown).
- Response View:** Pretty, Raw, Preview, Visualize, JSON (dropdown), and a copy icon.
- Response Body:**

```
1 {
2   "firstName": "Pierce",
3   "lastName": "Isaacson",
4   "orderNumber": 0,
5   "phoneNumber": "411-222-3333",
6   "id": 1
7 }
```


GET/Retrieval successful

PUT/Update of second entry of Chelsea earlier in PDF

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** localhost:8080/update/customer/
- Send Button:** A blue button labeled "Send" with a dropdown arrow.
- Tabs:** Params, Authorization, Headers (8), **Body** (selected), Pre-request Script, Tests, Settings.
- Body Type:** none, form-data, x-www-form-urlencoded, **raw** (selected), binary, GraphQL, JSON (dropdown).
- Body Content:** A large text area containing a JSON object.
- Status Bar:** 200 OK, 29 ms, 170 B, Save Response (dropdown).
- Response View:** Pretty, Raw, Preview, Visualize, Text (dropdown), and a copy icon.
- Response Body:**

```
1 {
2   "id": "3",
3   "firstName": "Jen",
4   "lastName": "Baker",
5   "orderNumber": "20",
6   "phoneNumber": "333-555-6666"
7 }
8
```
- Footer:** 1 Updated

customer  Enter a SQL expression to filter results (use Ctrl+Space)					
	customer_id	first_name	last_name	order_number	phone_number
1	1	Pierce	Isaacson	0	111-222-3333
2	2	Chelsea	Gesh	0	222-111-3333
3	3	Jen	Baker	20	333-555-6666

PUT/Update successful

DELETE of the first entry

DELETE
localhost:8080/delete/customer/1/
Send

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies


none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify

1

Body
Cookies
Headers (5)
Test Results
200 OK
229 ms
170 B
Save Response

Pretty
Raw
Preview
Visualize
Text

1 Deleted

customer  Enter a SQL expression to filter results (use Ctrl+Space)					
	customer_id	first_name	last_name	order_number	phone_number
1	2	Chelsea	Gesh	0	222-111-3333
2	3	Jen	Baker	20	333-555-6666

DELETE successful

Would-be Entity Relationship Diagram

