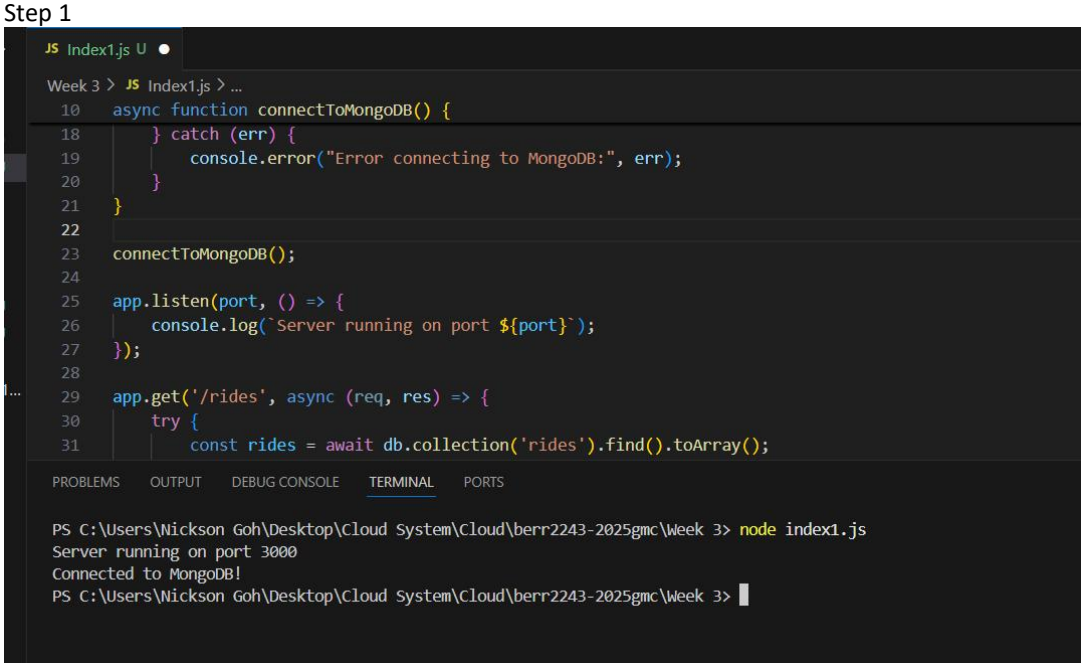
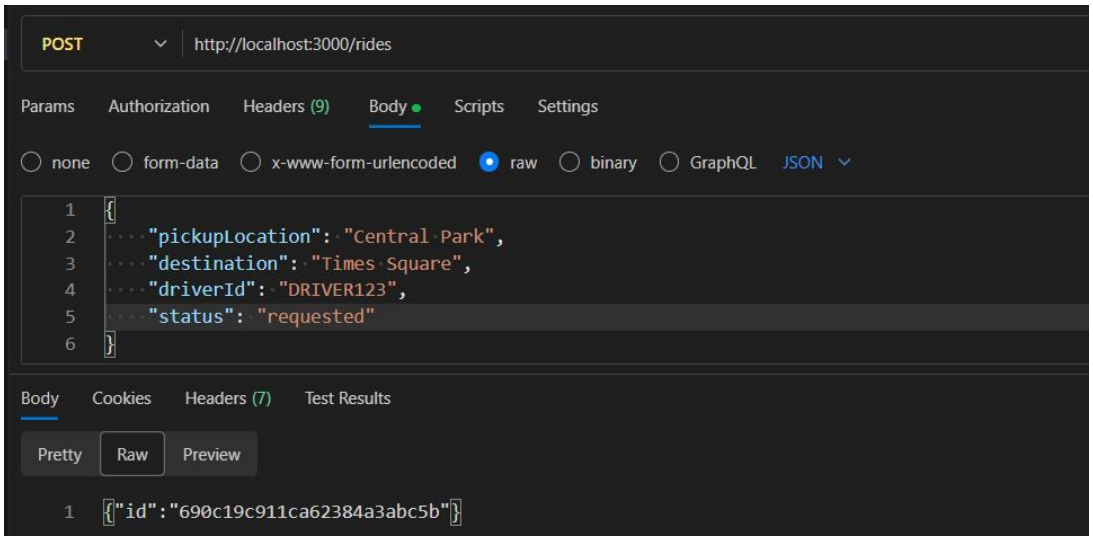
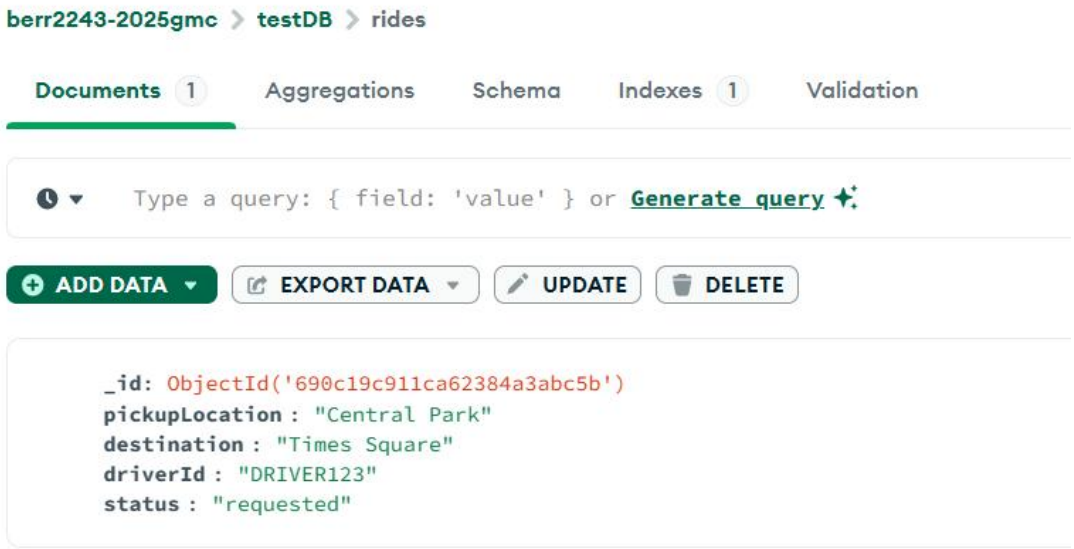
Step 1



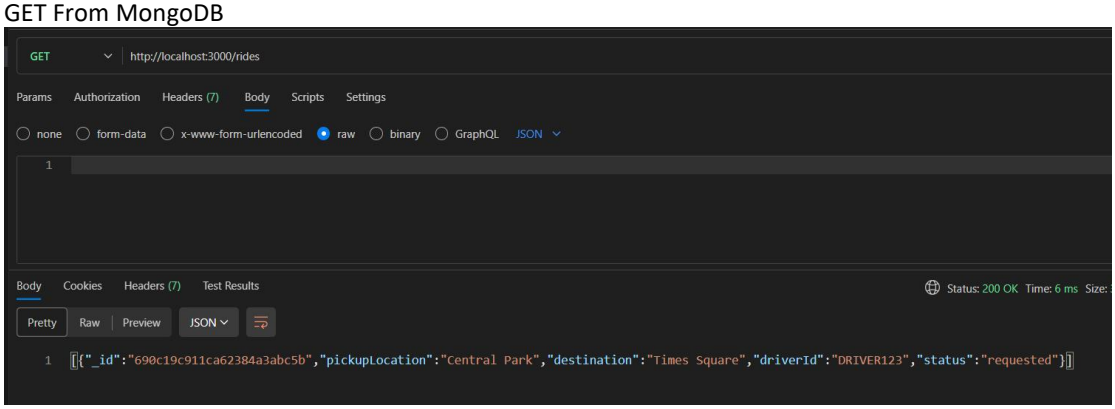
POST



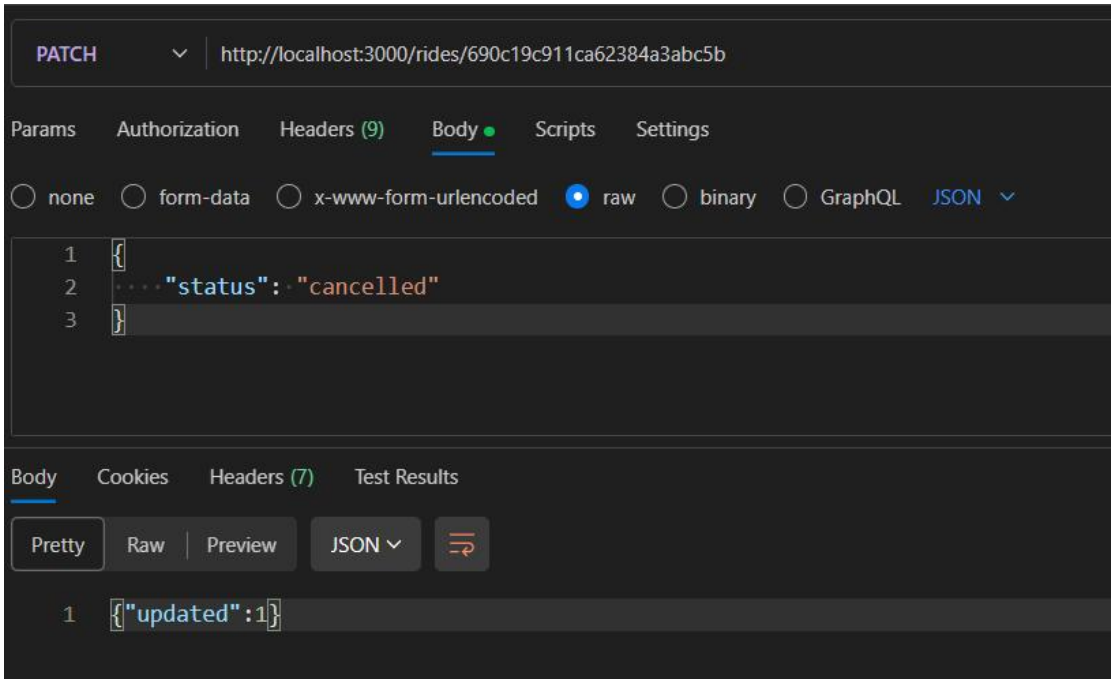
POST (Output)

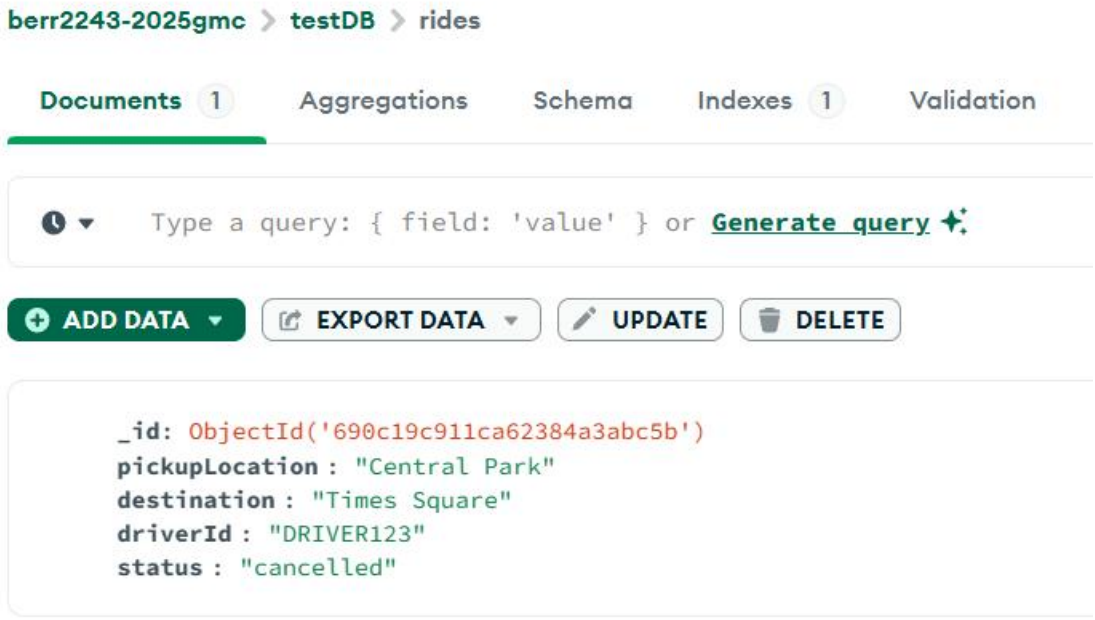


GET From MongoDB

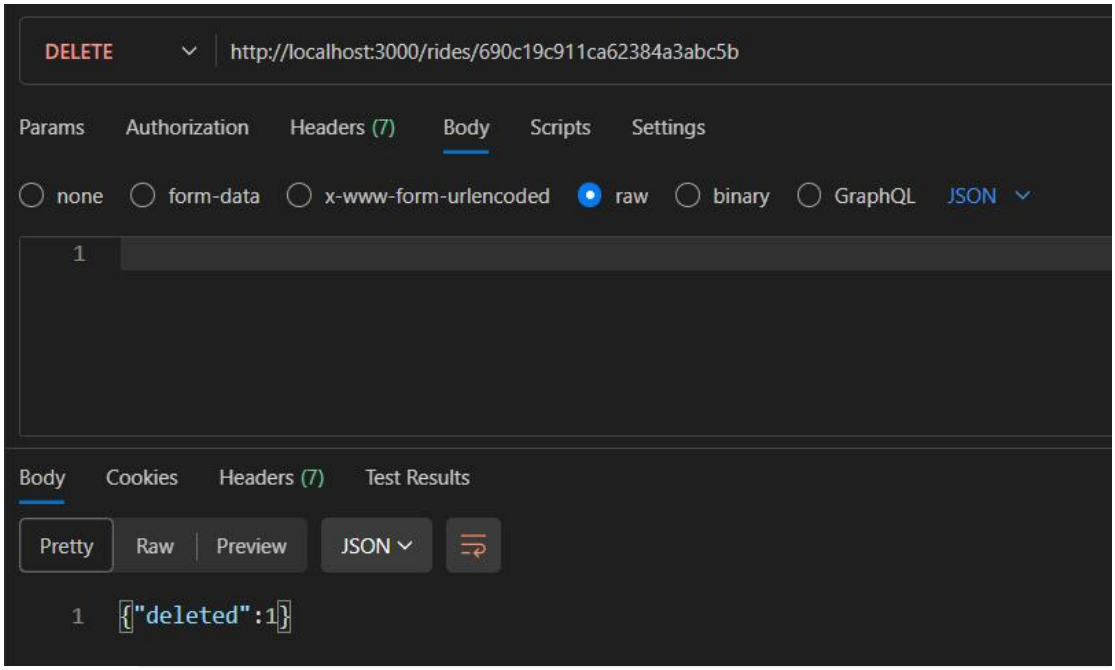


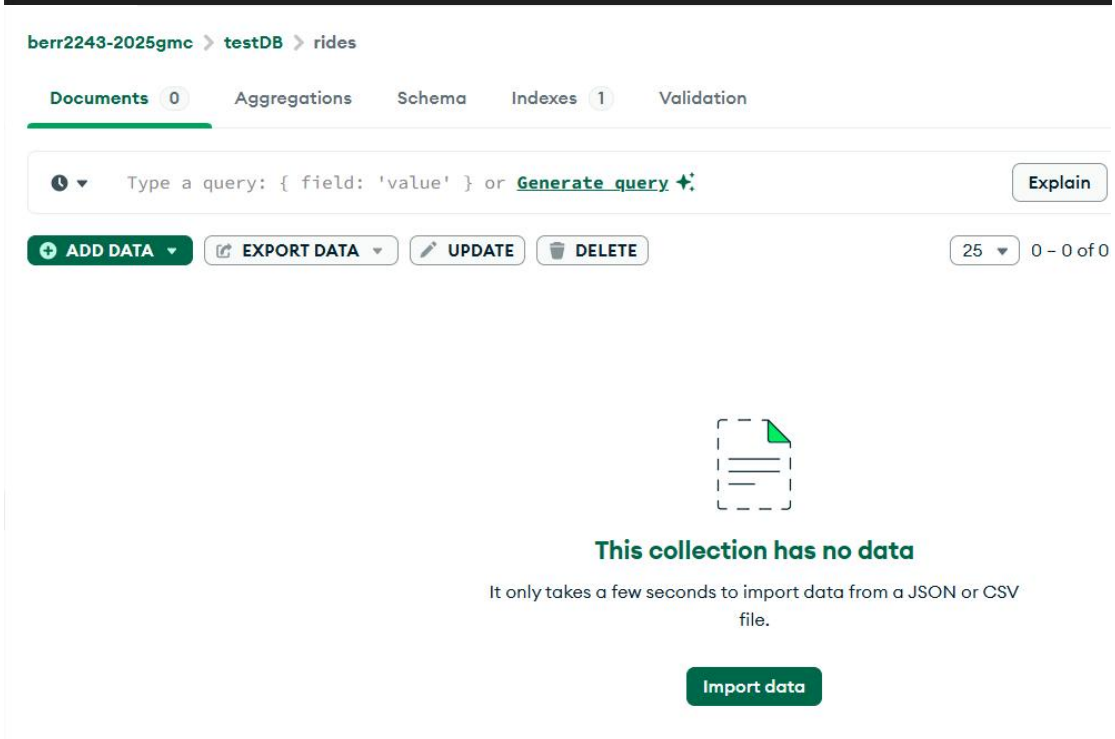
PATCH





DELETE


DELETE
http://localhost:3000/rides/690c19c911ca62384a3abc5b
Params
Authorization
Headers (7)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
JSON
1
{"deleted":1}


berr2243-2025gmc
testDB
rides
Documents 0
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
0 – 0 of 0
This collection has no data
It only takes a few seconds to import data from a JSON or CSV file.
Import data

Lab Questions
Answer by testing your API in Postman and observing responses.

1. POST Request:
o What HTTP status code is returned when a ride is created successfully?
=201 Created
o What is the structure of the response body?
=json({ id: result.insertedId })

2. GET Request:
o What happens if the rides collection is empty?
= **empty array ([])**

o What data type is returned in the response (array/object)?
=**array**

3. Fix PATCH and DELETE Error:
o Catch the error when requesting PATCH or DELETE API, then try to fix the issue reported.
=error (400 bad request)
To fix, add the ObjectId.

```
const { MongoClient, ObjectId } = require('mongodb');
```

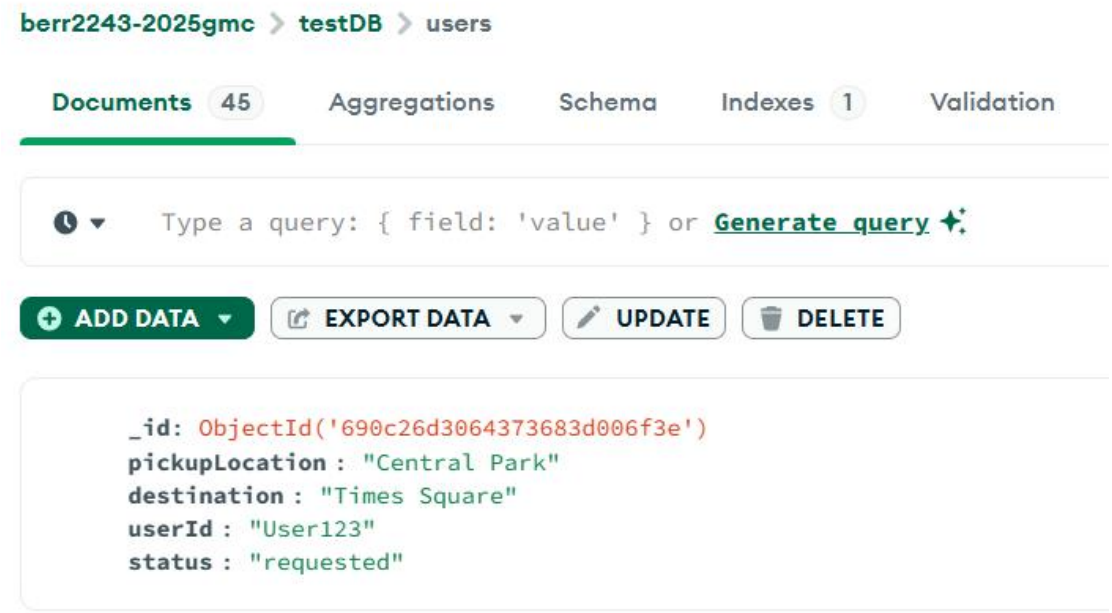
o If you try to update a non-existent ride ID, what status code is returned?
= **Invalid ride Id or data**

o What is the value of updated in the response if the update succeeds?
= **{"updated":1}**

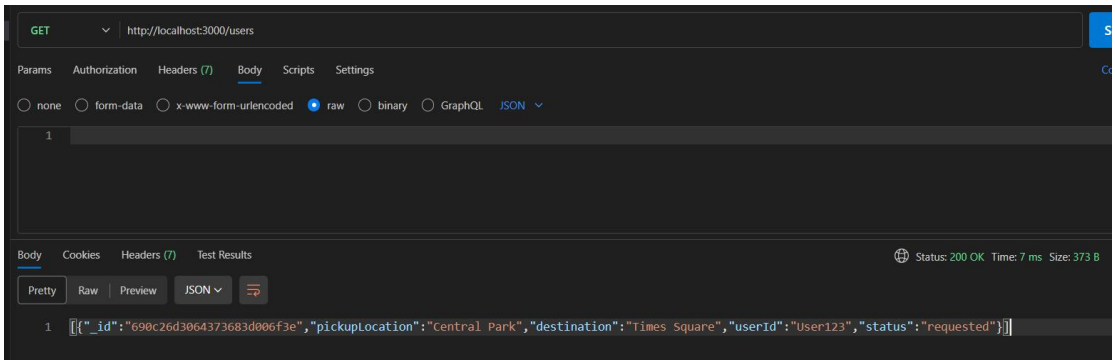
o How does the API differentiate between a successful deletion and a failed one?
= **if successful {"deleted":1} and status 200 OK if unsuccessful {"error":"Ride not found"} and status 404 not found.**

4. Users Endpoints:
o Based on the exercise above, create the endpoints to handle the CRUD operations for users account

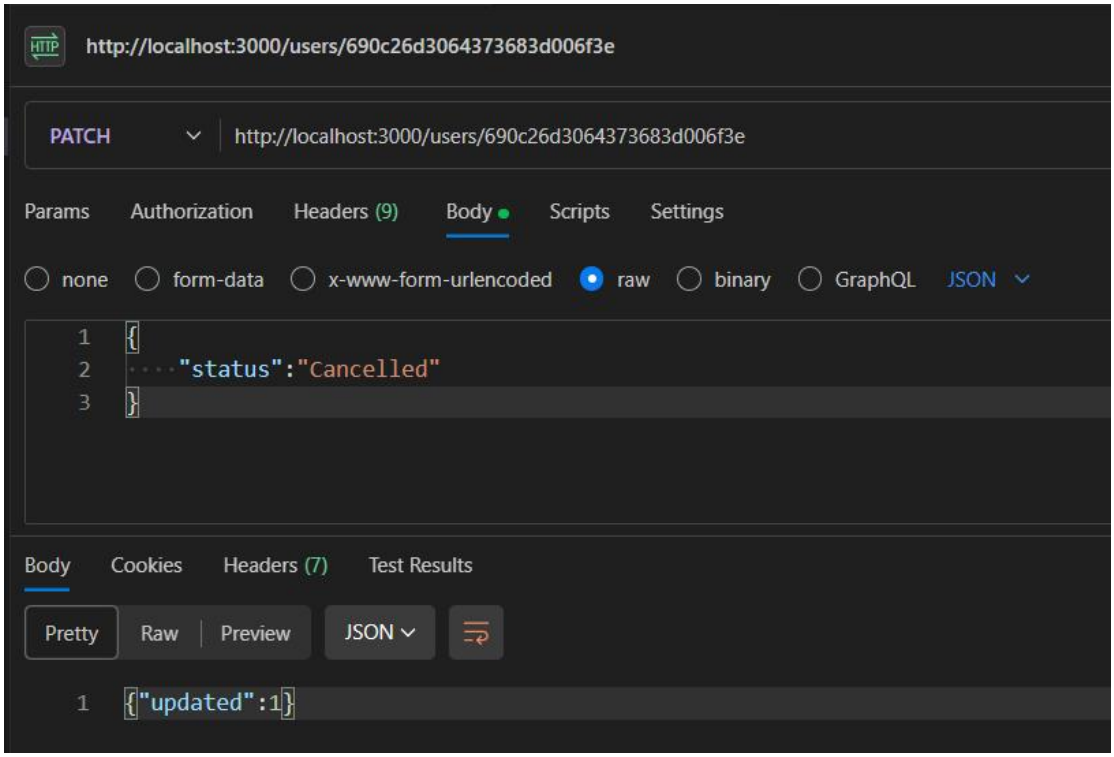
POST

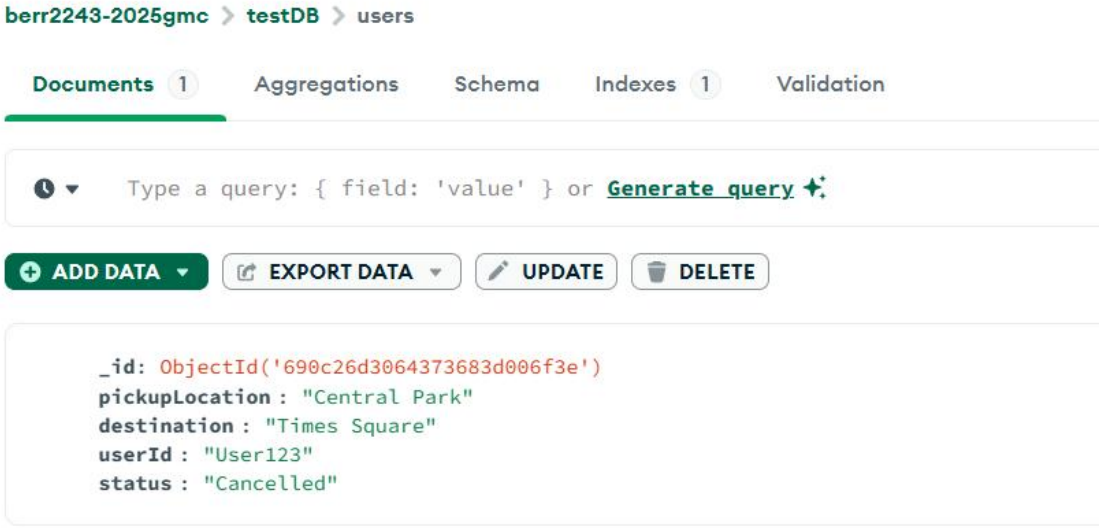


GET



PATCH


http://localhost:3000/users/690c26d3064373683d006f3e
PATCH
http://localhost:3000/users/690c26d3064373683d006f3e
Params
Authorization
Headers (9)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1 {
2 "status":"Cancelled"
3 }
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
JSON
1 {"updated":1}


berr2243-2025gmc
testDB
users
Documents 1
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
ADD DATA
EXPORT DATA
UPDATE
DELETE
_id: ObjectId('690c26d3064373683d006f3e')
pickupLocation : "Central Park"
destination : "Times Square"
userId : "User123"
status : "Cancelled"

5. FrontEnd:
o Upload the Postman JSON to any AI tools, and generate a simple HTML and JS Dashboard for you