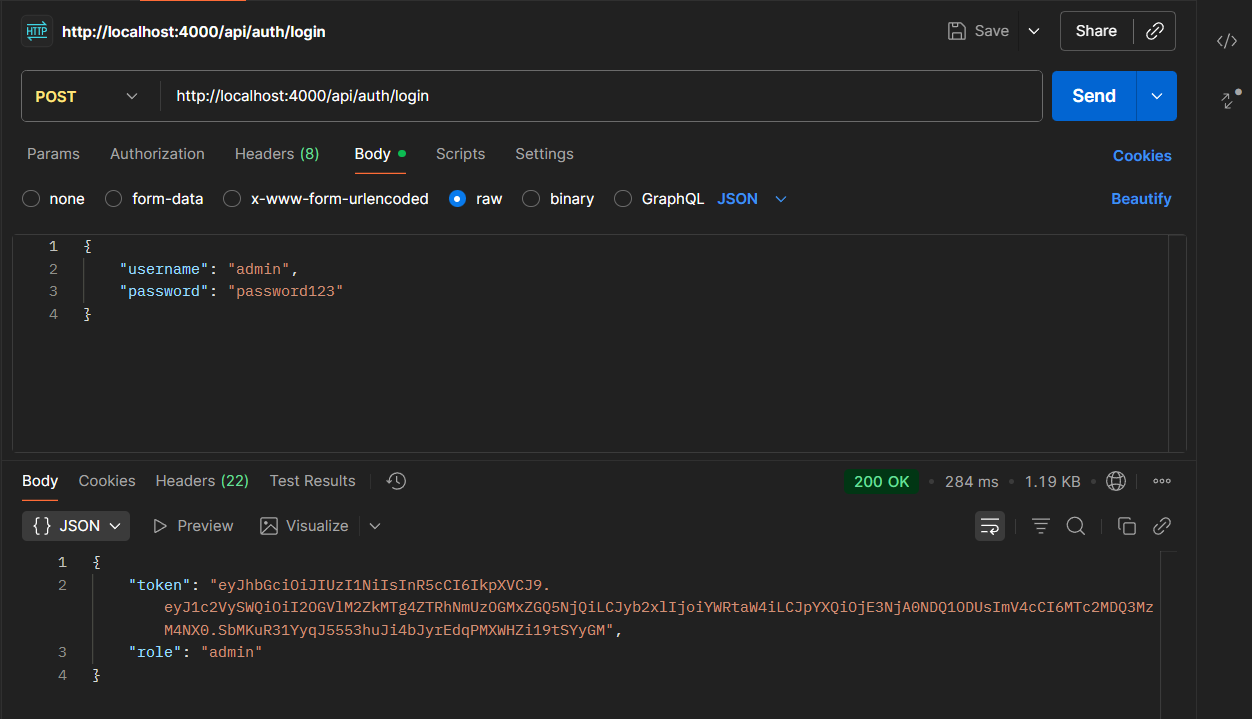
Unit test- Bus tracking system

1. POST /api/auth/login — returns JWT token

Request

* Method: POST
* URL: {{baseUrl}}/api/auth/login
* Headers: Content-Type: application/json
* Body (raw JSON):



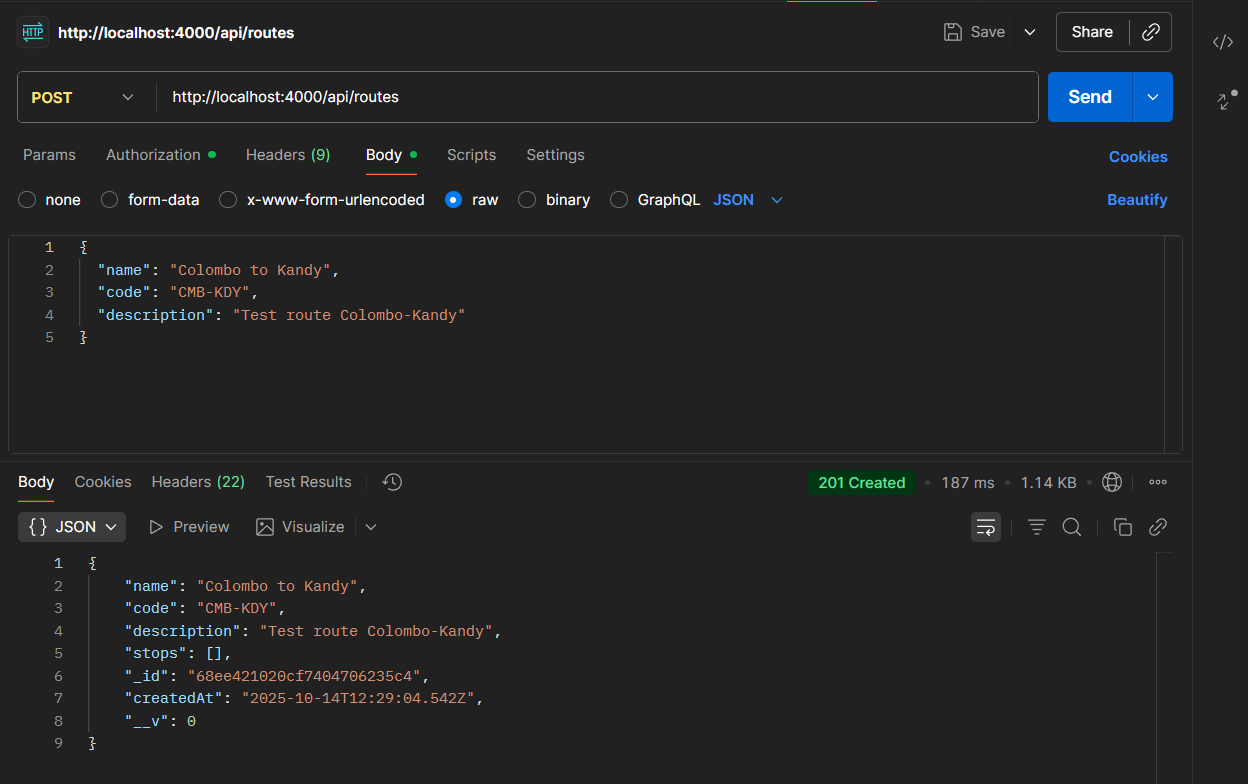
Expected

* Status: 200 OK
* Body: contains token string

1. POST /api/routes — create route (protected)

Request

* Method: POST
* URL: {{baseUrl}}/api/routes
* Headers:
  + Content-Type: application/json
  + Authorization: Bearer {{token}}
* Body



Expected

* Status: 201 created
* Body: route object with \_id, name, code, stops: []

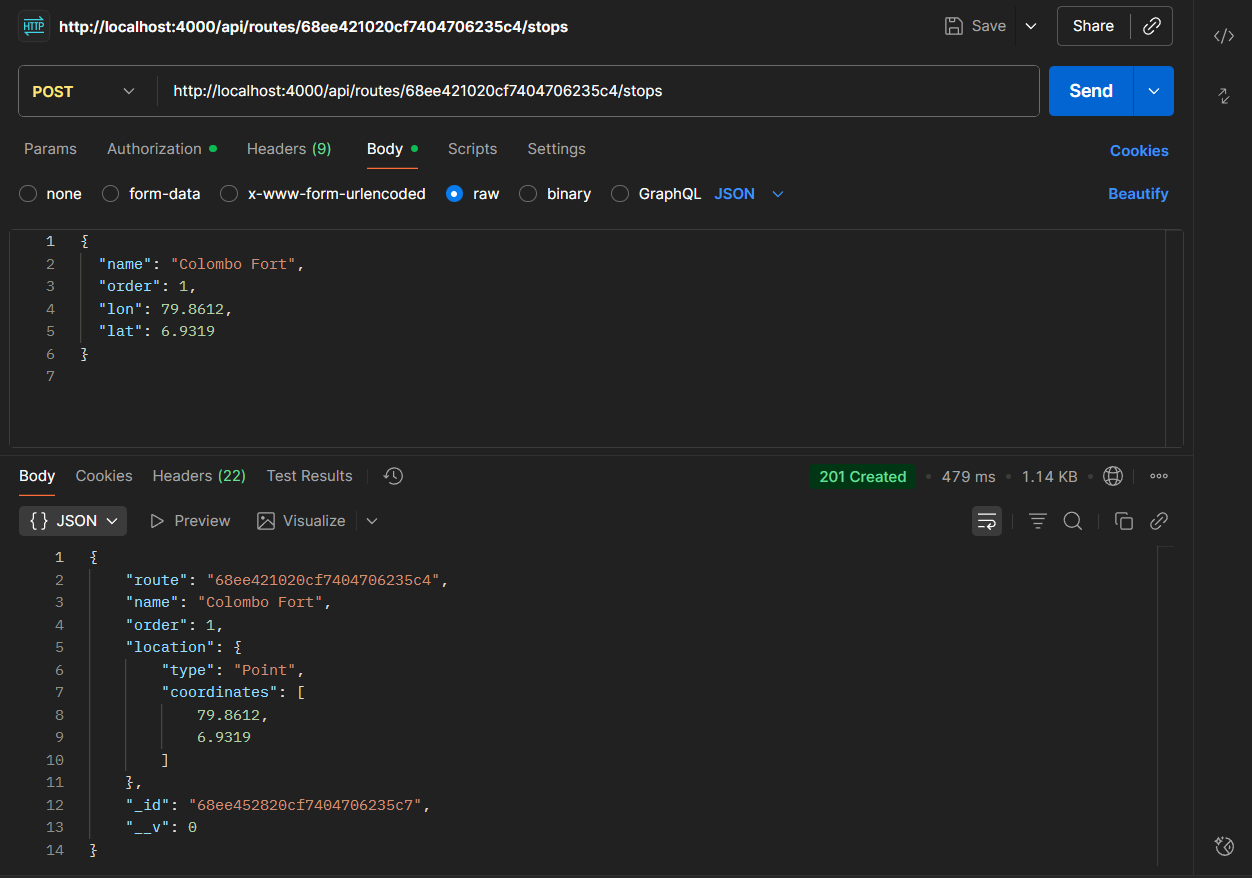


1. POST /api/routes/:routeId/stops — add stop (protected)

Request

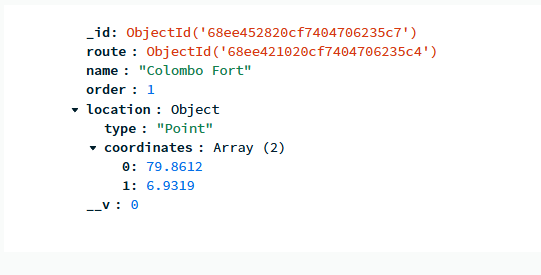
* Method: POST
* URL: {{baseUrl}}/api/routes/{{routeId}}/stops
* Headers:
  + Content-Type: application/json
* Authorization: Bearer {{token}}

Body



Expected

* Status 201 Created
* Body: stop object with \_id, route, location.coordinates



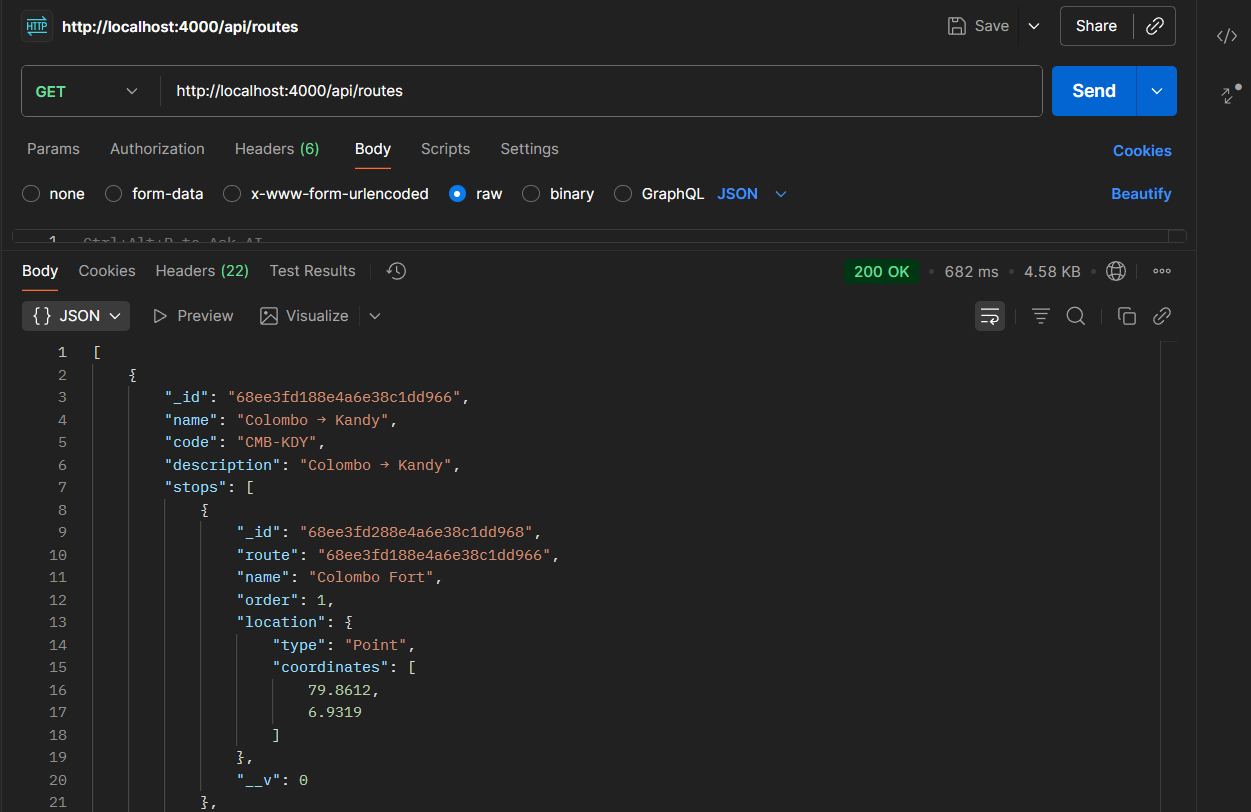
1. GET /api/routes — list routes

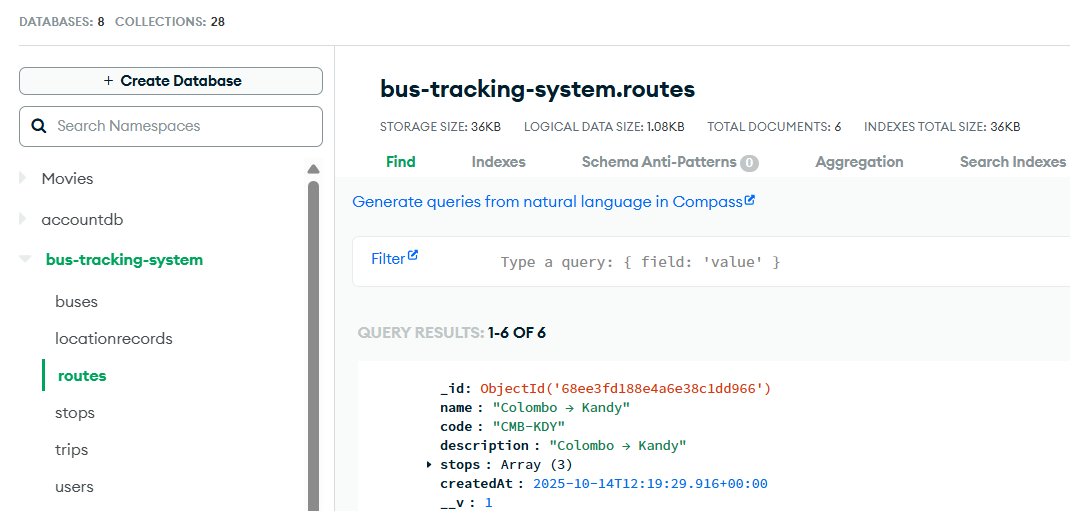
Request

* Method: GET
* URL: http://localhost:4000/api/routes
* Headers: Accept: application/json

Expected

* Status 200 OK

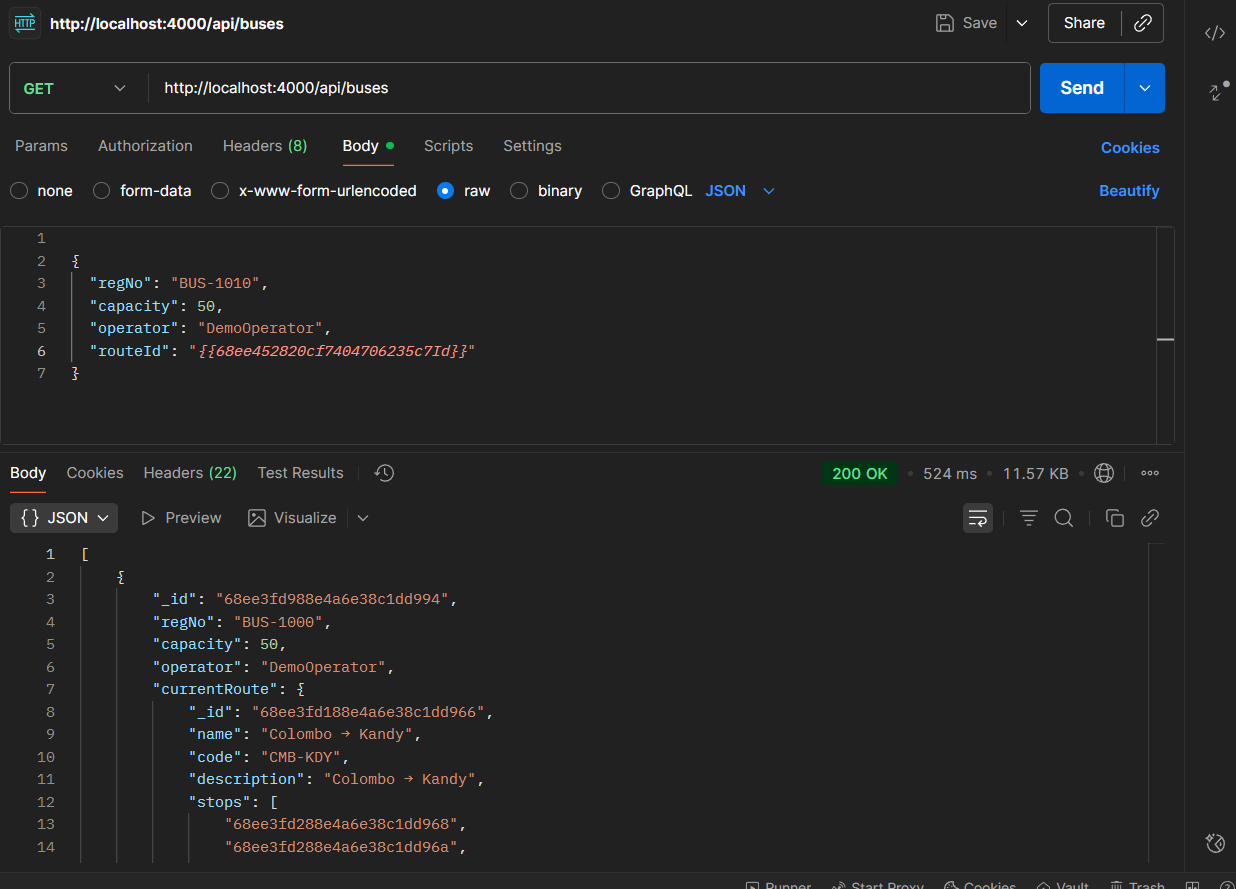


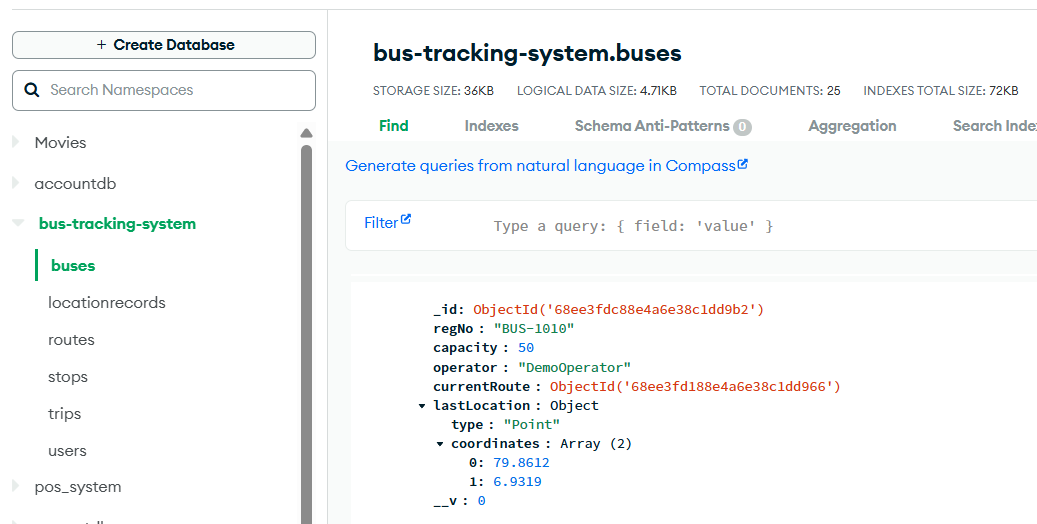


1. POST /api/buses — create bus (protected)

Request

* Method: POST
* URL: http://localhost:4000/api/buses
* Headers:
  + Content-Type: application/json
  + Authorization: Bearer {{token}}





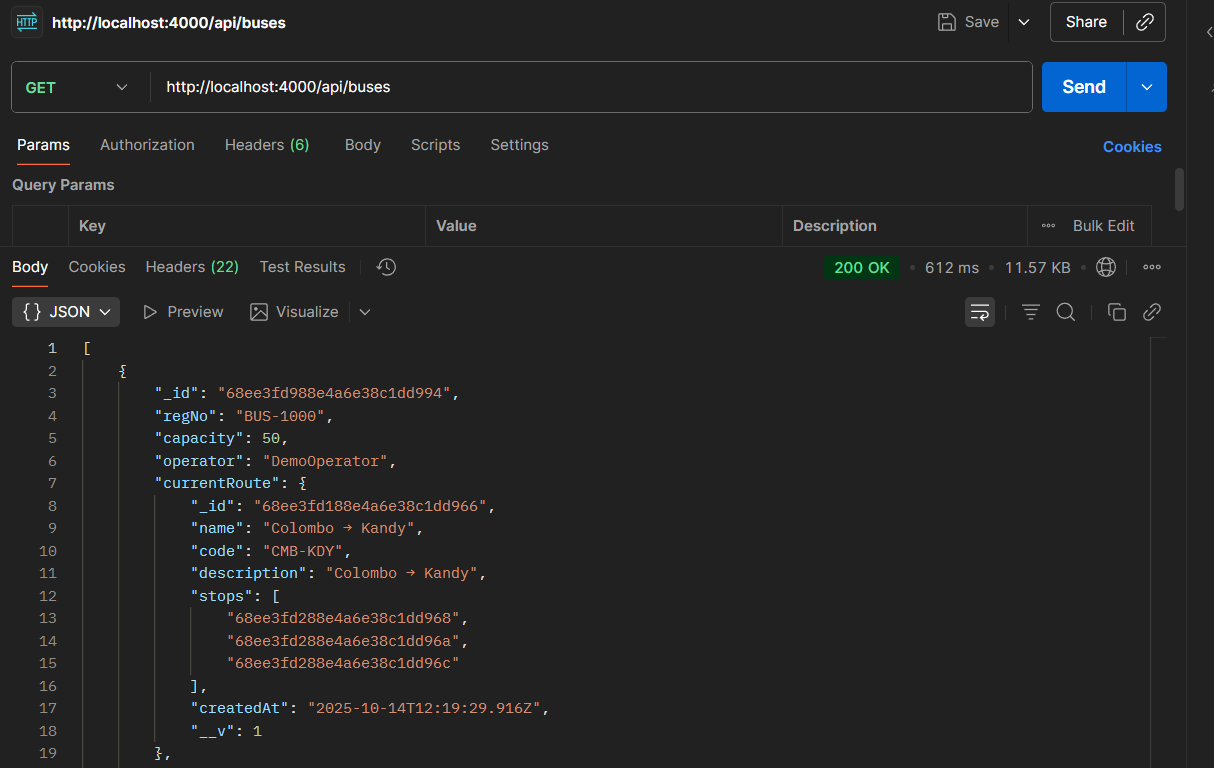
1. GET /api/buses — list buses

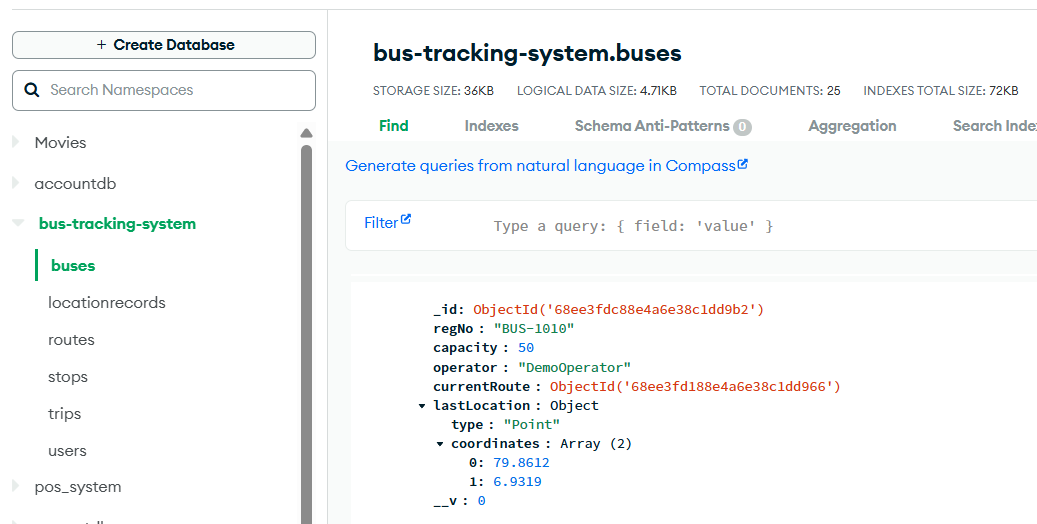
Request

* Method: GET
* URL: <http://localhost:4000/api/buses>
* Headers: Accept: application/json

Expected

* Status 200 OK
* Body: array containing the bus with \_id == {{busId}}





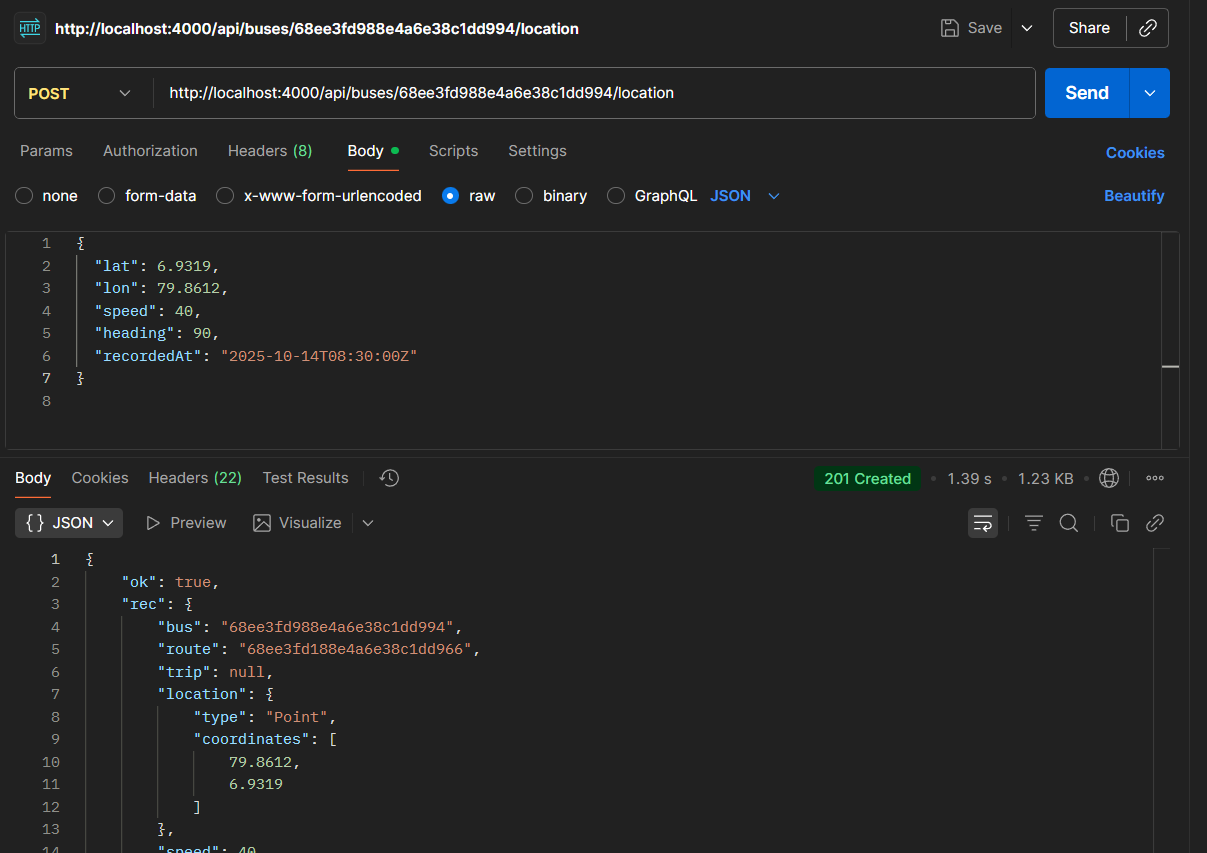
1. POST /api/buses/:id/location — ingest location

Request

* Method: POST
* URL: {{baseUrl}}/api/buses/{{busId}}/location

Headers:

* Content-Type: application/json



1. GET /api/map/active-buses — latest known locations

Request

* Method: GET
* URL: {{baseUrl}}/api/map/active-buses
* Headers: Accept: application/json

Expected

* Status 200

