WEB PROGRAMMING PROJECT

DOCUMENTATION

Team Members

Gokul Sreekumar B180575CS

Jessiya Joy B180462CS

Anna Susan Cherian B180552CS

Submitted on: 16th November 2021

Project GitHub: gokulsreekumar/LibraryAPI-using-Ruby-on-Rails (github.com)

API DOCUMENTATION

OUTLINE

Base URL : http://localhost:3000
(Run the rails application in localhost on port 3000)

GET	/api/v1/books	Find all books
GET	/api/v1/books/{book_id}	Find book by ID
POST	/api/v1/books	Add a new book
PUT	/api/v1/books/{book_id}	Update an existing book
DELETE	/api/v1/books/{book_id}	Delete a book

GET /api/v1/books

Description:

Fetch the list of all books currently present in the database.

Responses:

200 Successful Operation

The response body contains all the books in the database in JSON format.

GET /api/v1/books/{book_id}

Parameters:

book_id : book_id of the book to fetch

Description:

To fetch details of book based on the given book_id

Responses:

200 **Successful Operation**

The response body contains the details of the book associated with the book_id in JSON format.

400 **Invalid book_id / Book not found**

The response body contains a message "Unable to get the Book."

POST /api/v1/books

Description:

To create an entry of a book in the database.

Responses:

200 **Successful Operation**

The response body contains the details of the book just submitted along with the three new fields - id and timestamps of creation and last update.

400 **Unsuccessful book creation**

The response body contains a message "Unable to create Book."

PUT /api/v1/books/{book_id}

Parameters:

book_id : book_id of the book to fetch

Description:

To edit the details of the book associated with the book_id.

Responses:

200 **Successful Operation**

The response body contains a message "Book updated successfully"

400 **Invalid book_id / Book not found**

The response body contains a message "Unable to update Book."

DELETE /api/v1/books/{book_id}

Parameters:

book_id : book_id of the book to fetch

Description:

To delete an entry of a book from the database using book ID.

Responses:

200 Successful Operation

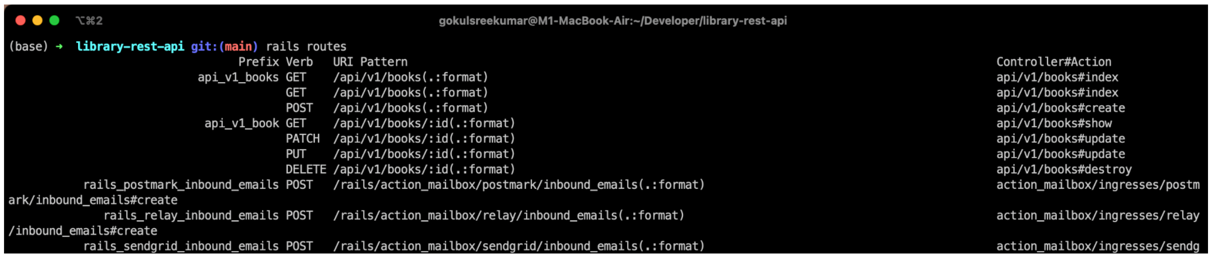
The response body contains a message "Book deleted successfully".

400 Invalid book_id / Book not found

The response body contains a message "Unable to delete Book."

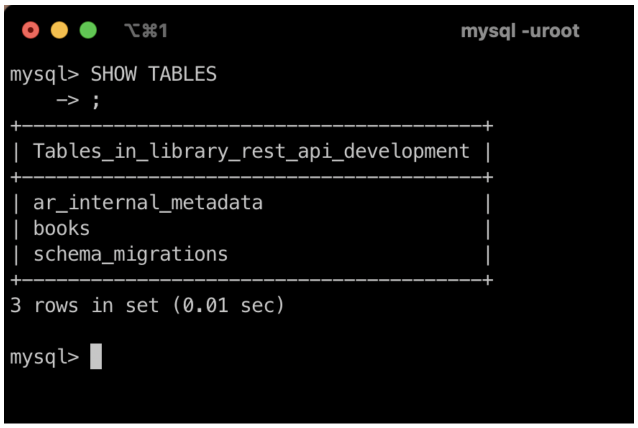
RAILS API SERVER CHECKS

Check Controller Routes for the API



```
⌥⌘2                                   gokulsreekumar@M1-MacBook-Air:~/Developer/library-rest-api
(base) ➜  library-rest-api git:(main) rails routes
                                  Prefix Verb   URI Pattern                                                                                       Controller#Action
                            api_v1_books GET    /api/v1/books(.:format)                                                                           api/v1/books#index
                                         GET    /api/v1/books(.:format)                                                                           api/v1/books#index
                                         POST   /api/v1/books(.:format)                                                                           api/v1/books#create
                             api_v1_book GET    /api/v1/books/:id(.:format)                                                                       api/v1/books#show
                                         PATCH  /api/v1/books/:id(.:format)                                                                       api/v1/books#update
                                         PUT    /api/v1/books/:id(.:format)                                                                       api/v1/books#update
                                         DELETE /api/v1/books/:id(.:format)                                                                       api/v1/books#destroy
           rails_postmark_inbound_emails POST   /rails/action_mailbox/postmark/inbound_emails(.:format)                                           action_mailbox/ingresses/postm
ark/inbound_emails#create
              rails_relay_inbound_emails POST   /rails/action_mailbox/relay/inbound_emails(.:format)                                              action_mailbox/ingresses/relay
/inbound_emails#create
           rails_sendgrid_inbound_emails POST   /rails/action_mailbox/sendgrid/inbound_emails(.:format)                                           action_mailbox/ingresses/sendg
```

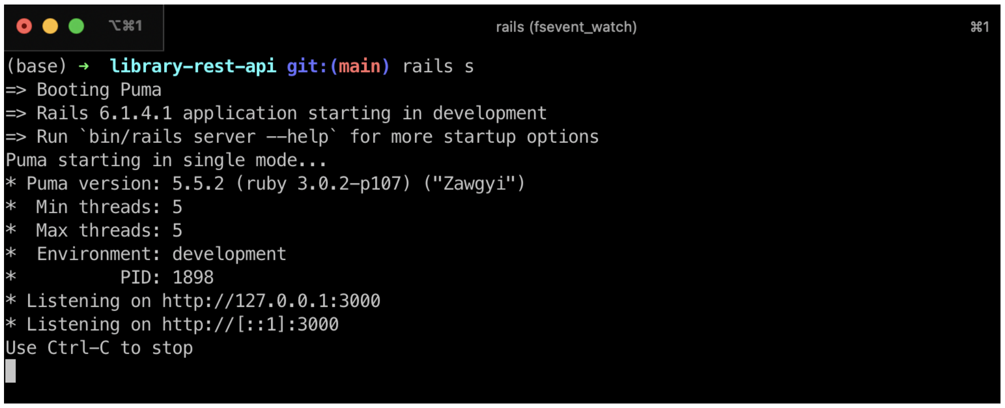
Check Existence of required tables in the database.



```
mysql> SHOW TABLES
    -> ;
+--------------------------------------+
| Tables_in_library_rest_api_development |
+--------------------------------------+
| ar_internal_metadata                 |
| books                                |
| schema_migrations                    |
+--------------------------------------+
3 rows in set (0.01 sec)

mysql> 
```

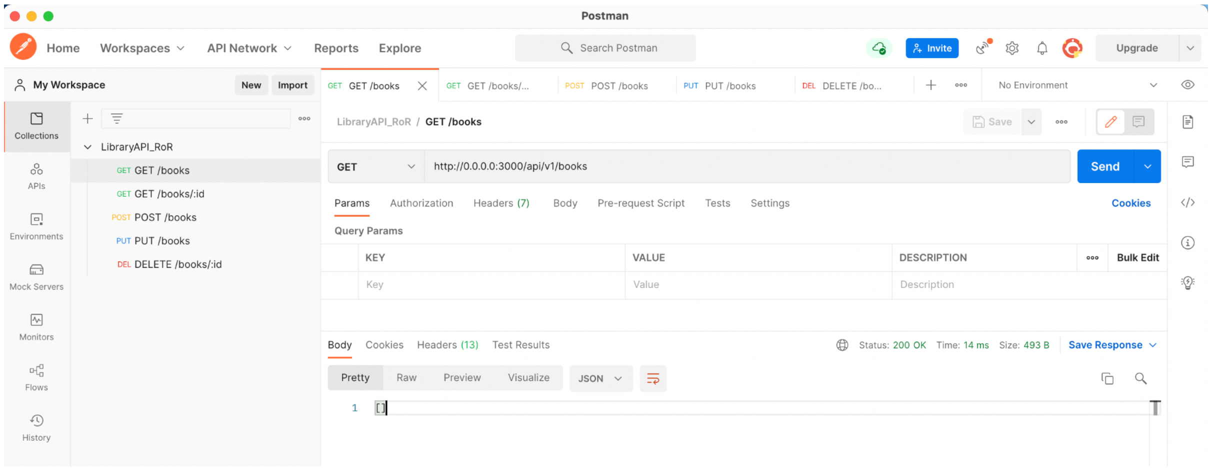
Run Rails Library API server



```
(base) ➜  library-rest-api git:(main) rails s
=> Booting Puma
=> Rails 6.1.4.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.5.2 (ruby 3.0.2-p107) ("Zawgyi")
*  Min threads: 5
*  Max threads: 5
*  Environment: development
*          PID: 1898
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
```

API TESTING USING POSTMAN

For testing, create Postman requests for all the **CRUD** (Create, Read, Update and Delete) operations on Book, namely - GET(single book and all books), POST, PUT, DELETE as follows:



LibraryAPI_RoR / GET /books

GET http://0.0.0.0:3000/api/v1/books

KEY	VALUE	DESCRIPTION
Key	Value	Description

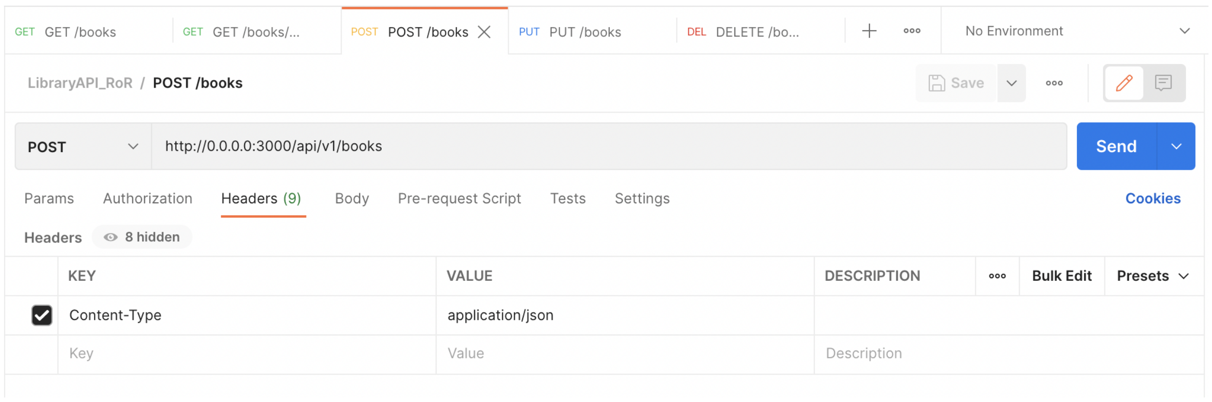
Status: 200 OK Time: 14 ms Size: 493 B

```
[]
```

Postman Collection of APIs

POST

Set Header as below:

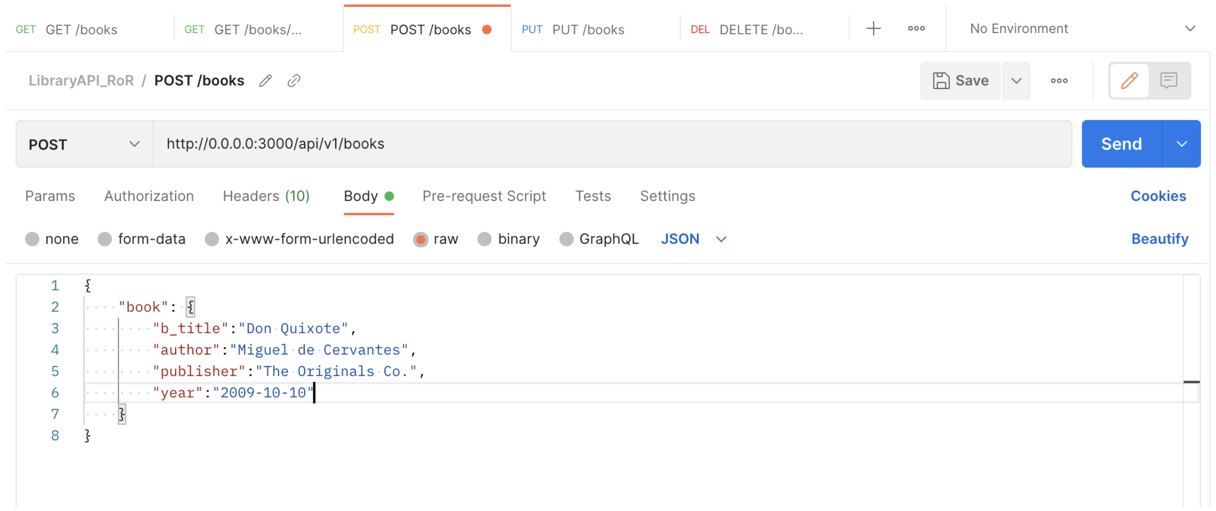


LibraryAPI_RoR / POST /books

POST http://0.0.0.0:3000/api/v1/books

	KEY	VALUE	DESCRIPTION
✓	Content-Type	application/json	
	Key	Value	Description

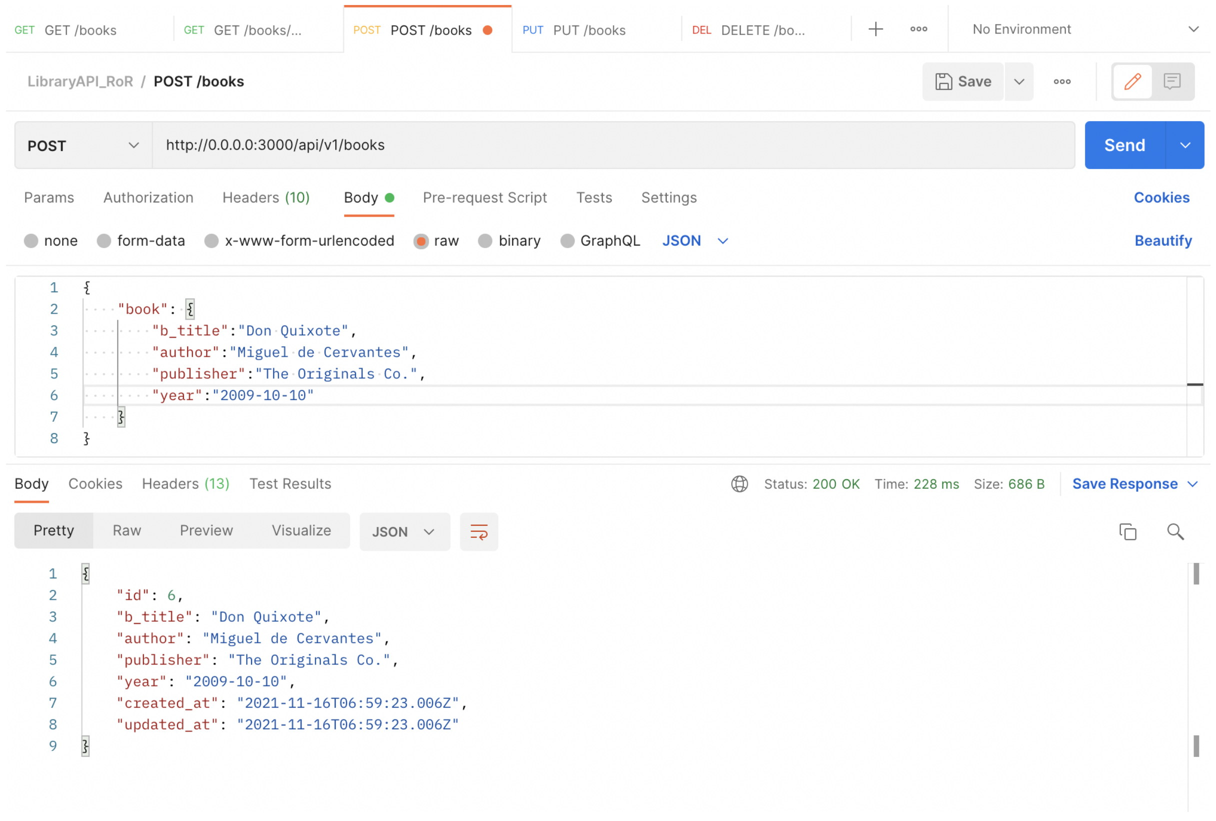
Body - JSON of the new book to be created:



POST http://0.0.0.0:3000/api/v1/books

```
{
    "book": {
        "b_title":"Don Quixote",
        "author":"Miguel de Cervantes",
        "publisher":"The Originals Co.",
        "year":"2009-10-10"
    }
}
```

Response JSON - the new Book that got created:



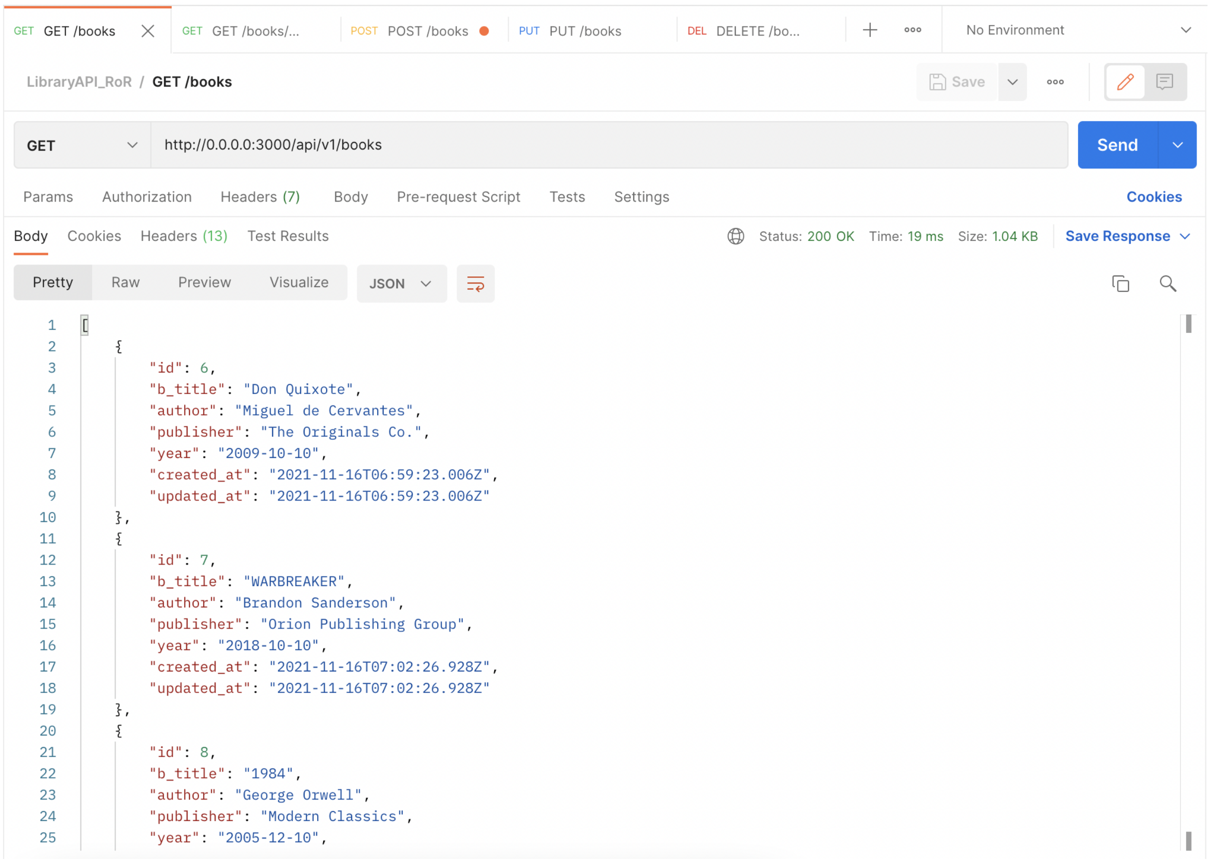
POST http://0.0.0.0:3000/api/v1/books

```
{
    "book": {
        "b_title":"Don Quixote",
        "author":"Miguel de Cervantes",
        "publisher":"The Originals Co.",
        "year":"2009-10-10"
    }
}
```

Status: 200 OK Time: 228 ms Size: 686 B

```
{
    "id": 6,
    "b_title": "Don Quixote",
    "author": "Miguel de Cervantes",
    "publisher": "The Originals Co.",
    "year": "2009-10-10",
    "created_at": "2021-11-16T06:59:23.006Z",
    "updated_at": "2021-11-16T06:59:23.006Z"
}
```

GET List of Books

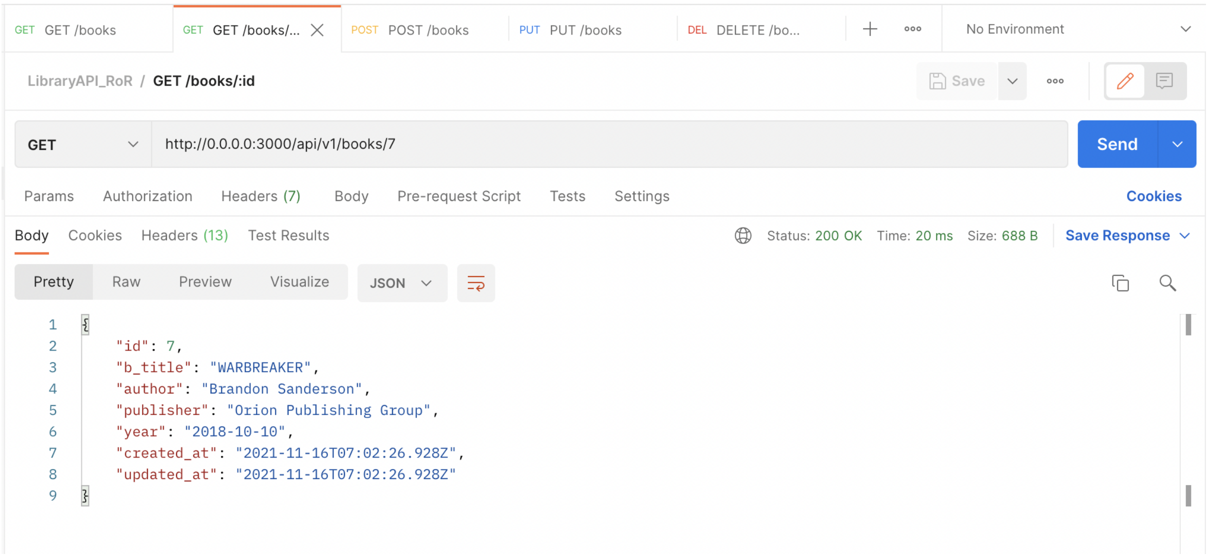


```
GET http://0.0.0.0:3000/api/v1/books
```

Status: 200 OK Time: 19 ms Size: 1.04 KB

```
[
    {
        "id": 6,
        "b_title": "Don Quixote",
        "author": "Miguel de Cervantes",
        "publisher": "The Originals Co.",
        "year": "2009-10-10",
        "created_at": "2021-11-16T06:59:23.006Z",
        "updated_at": "2021-11-16T06:59:23.006Z"
    },
    {
        "id": 7,
        "b_title": "WARBREAKER",
        "author": "Brandon Sanderson",
        "publisher": "Orion Publishing Group",
        "year": "2018-10-10",
        "created_at": "2021-11-16T07:02:26.928Z",
        "updated_at": "2021-11-16T07:02:26.928Z"
    },
    {
        "id": 8,
        "b_title": "1984",
        "author": "George Orwell",
        "publisher": "Modern Classics",
        "year": "2005-12-10",
```

GET a Book by ID

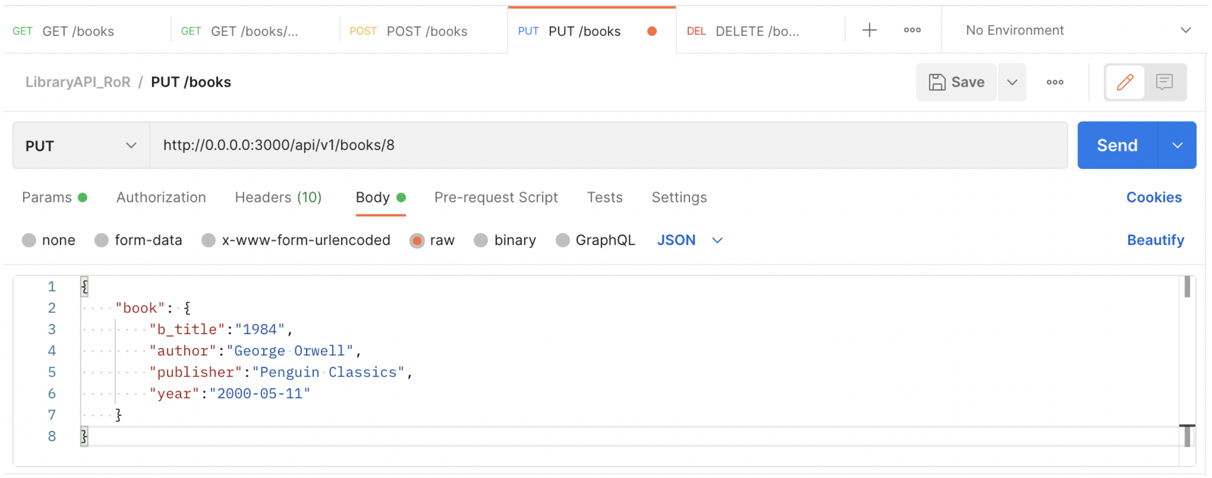


```
GET http://0.0.0.0:3000/api/v1/books/7
```

Status: 200 OK Time: 20 ms Size: 688 B

```
{
    "id": 7,
    "b_title": "WARBREAKER",
    "author": "Brandon Sanderson",
    "publisher": "Orion Publishing Group",
    "year": "2018-10-10",
    "created_at": "2021-11-16T07:02:26.928Z",
    "updated_at": "2021-11-16T07:02:26.928Z"
}
```

PUT (Edit a book: Here book with id=8 modified with new publisher and year)

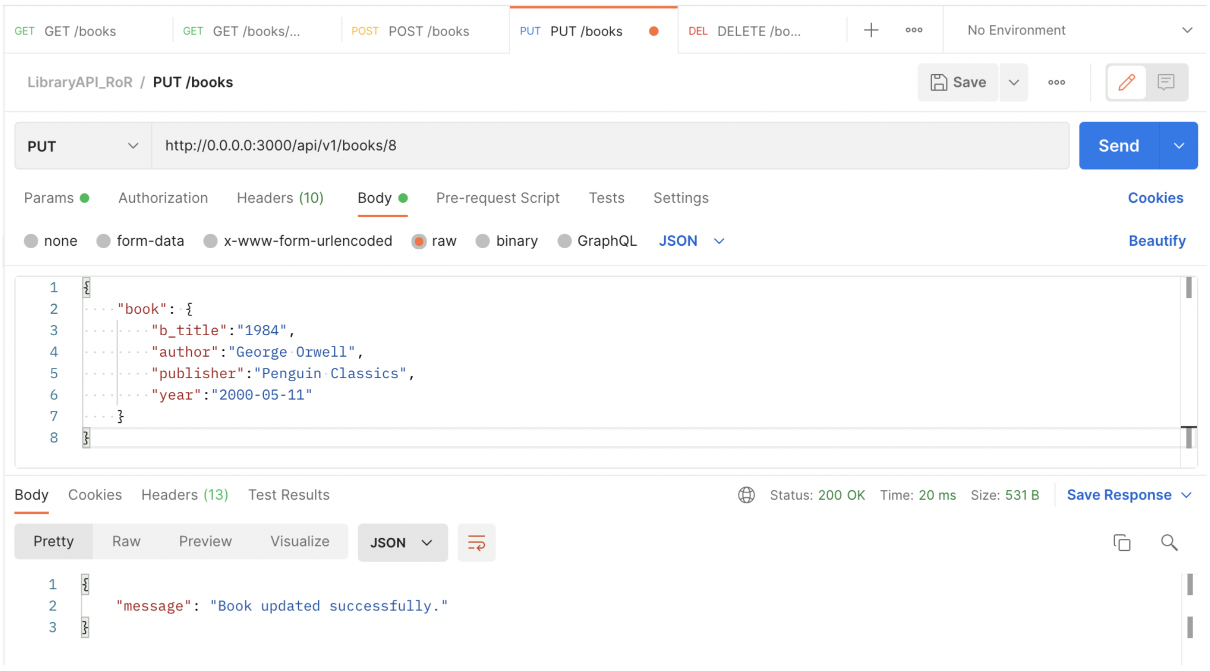


PUT http://0.0.0.0:3000/api/v1/books/8

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```
{
    "book": {
        "b_title":"1984",
        "author":"George Orwell",
        "publisher":"Penguin Classics",
        "year":"2000-05-11"
    }
}
```

Response JSON for PUT



PUT http://0.0.0.0:3000/api/v1/books/8

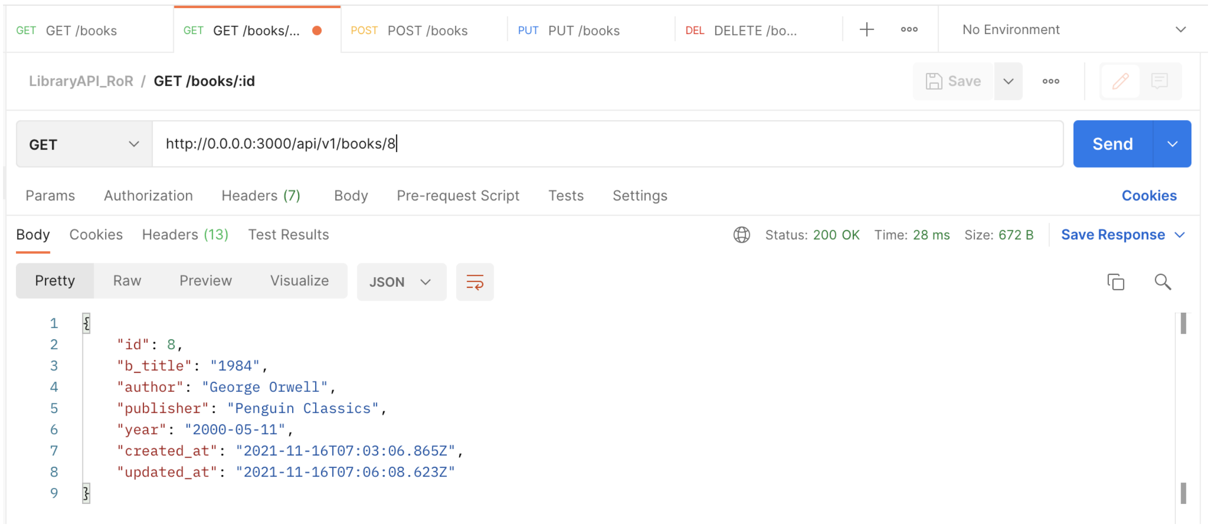
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```
{
    "book": {
        "b_title":"1984",
        "author":"George Orwell",
        "publisher":"Penguin Classics",
        "year":"2000-05-11"
    }
}
```

Status: 200 OK Time: 20 ms Size: 531 B

```
{
    "message": "Book updated successfully."
}
```

Check the edited book:

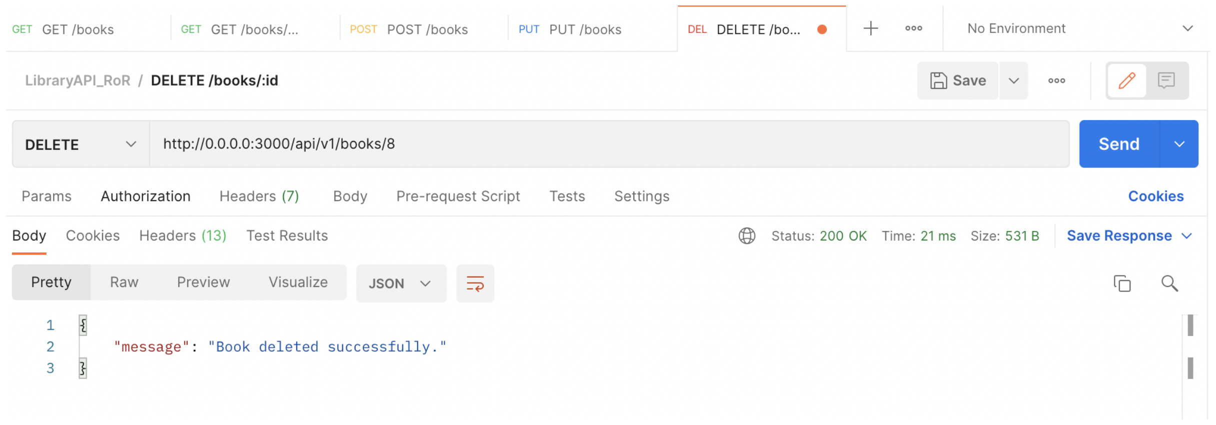


http://0.0.0.0:3000/api/v1/books/8

Status: 200 OK Time: 28 ms Size: 672 B

```
{
    "id": 8,
    "b_title": "1984",
    "author": "George Orwell",
    "publisher": "Penguin Classics",
    "year": "2000-05-11",
    "created_at": "2021-11-16T07:03:06.865Z",
    "updated_at": "2021-11-16T07:06:08.623Z"
}
```

DELETE a book by id

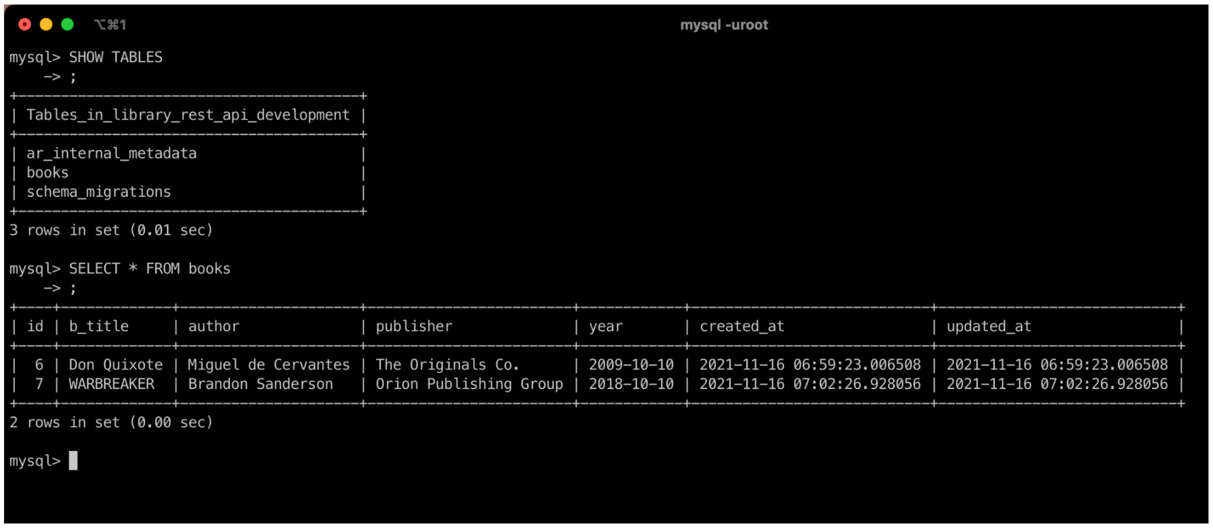


http://0.0.0.0:3000/api/v1/books/8

Status: 200 OK Time: 21 ms Size: 531 B

```
{
    "message": "Book deleted successfully."
}
```

Verify Final State of the Database:



mysql -uroot

```
mysql> SHOW TABLES
    -> ;
+----------------------------------------+
| Tables_in_library_rest_api_development |
+----------------------------------------+
| ar_internal_metadata                   |
| books                                  |
| schema_migrations                      |
+----------------------------------------+
3 rows in set (0.01 sec)

mysql> SELECT * FROM books
    -> ;
+----+-------------+---------------------+-----------------------+------------+----------------------------+----------------------------+
| id | b_title     | author              | publisher             | year       | created_at                 | updated_at                 |
+----+-------------+---------------------+-----------------------+------------+----------------------------+----------------------------+
|  6 | Don Quixote | Miguel de Cervantes | The Originals Co.     | 2009-10-10 | 2021-11-16 06:59:23.006508 | 2021-11-16 06:59:23.006508 |
|  7 | WARBREAKER  | Brandon Sanderson   | Orion Publishing Group | 2018-10-10 | 2021-11-16 07:02:26.928056 | 2021-11-16 07:02:26.928056 |
+----+-------------+---------------------+-----------------------+------------+----------------------------+----------------------------+
2 rows in set (0.00 sec)

mysql> 
```