

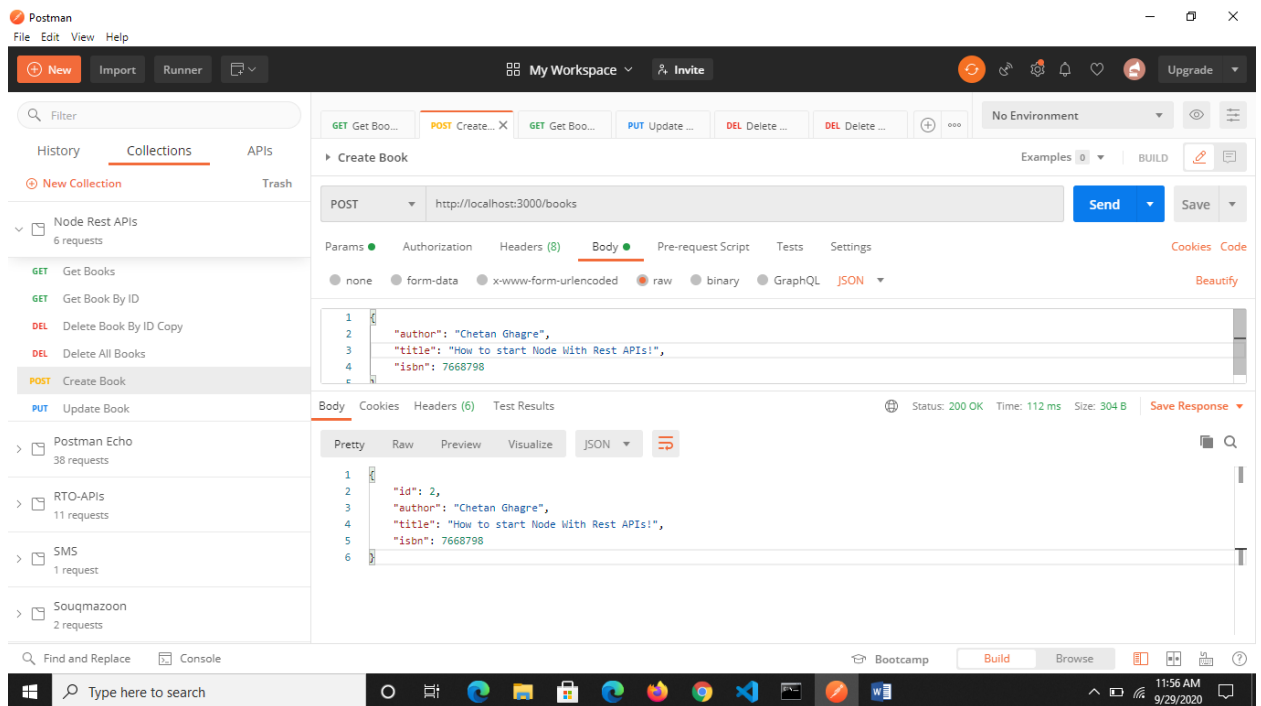
Test the APIs

Run our Node.js application with command: `node server.js`.
The console shows:

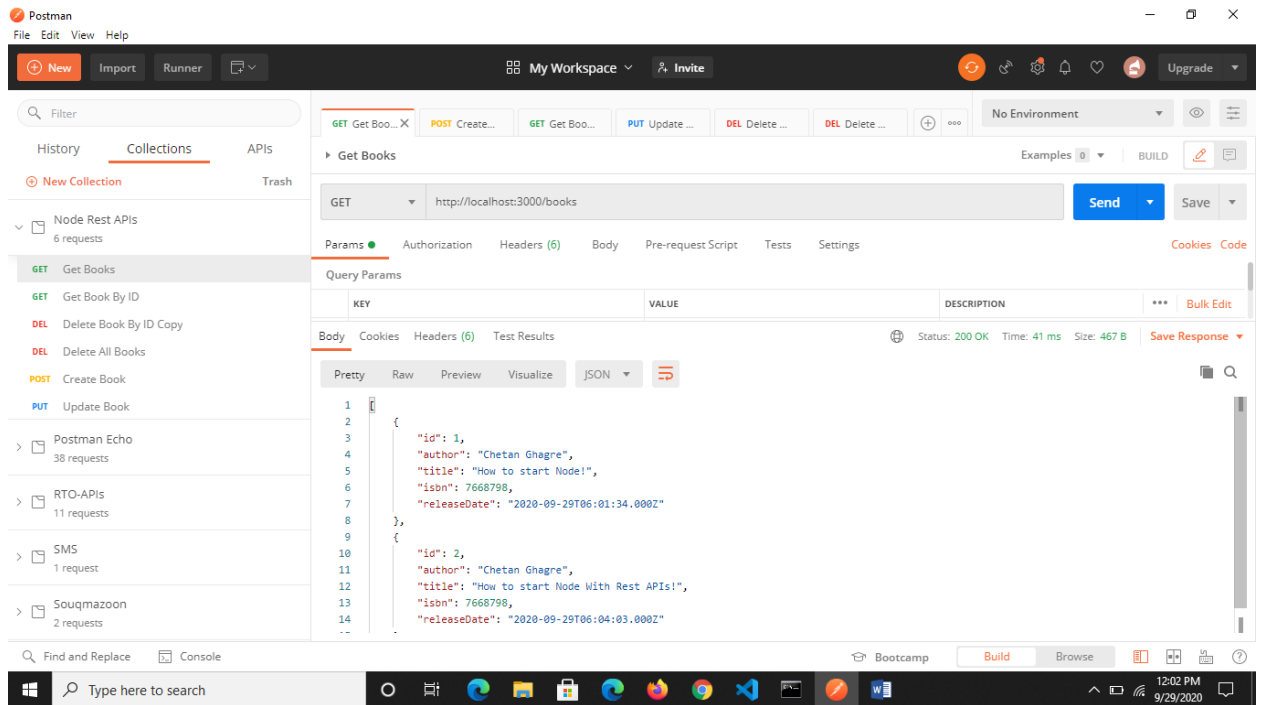
```
Server is running on port 3000.  
Successfully connected to the database.
```

Using Postman, we will test all the APIs.

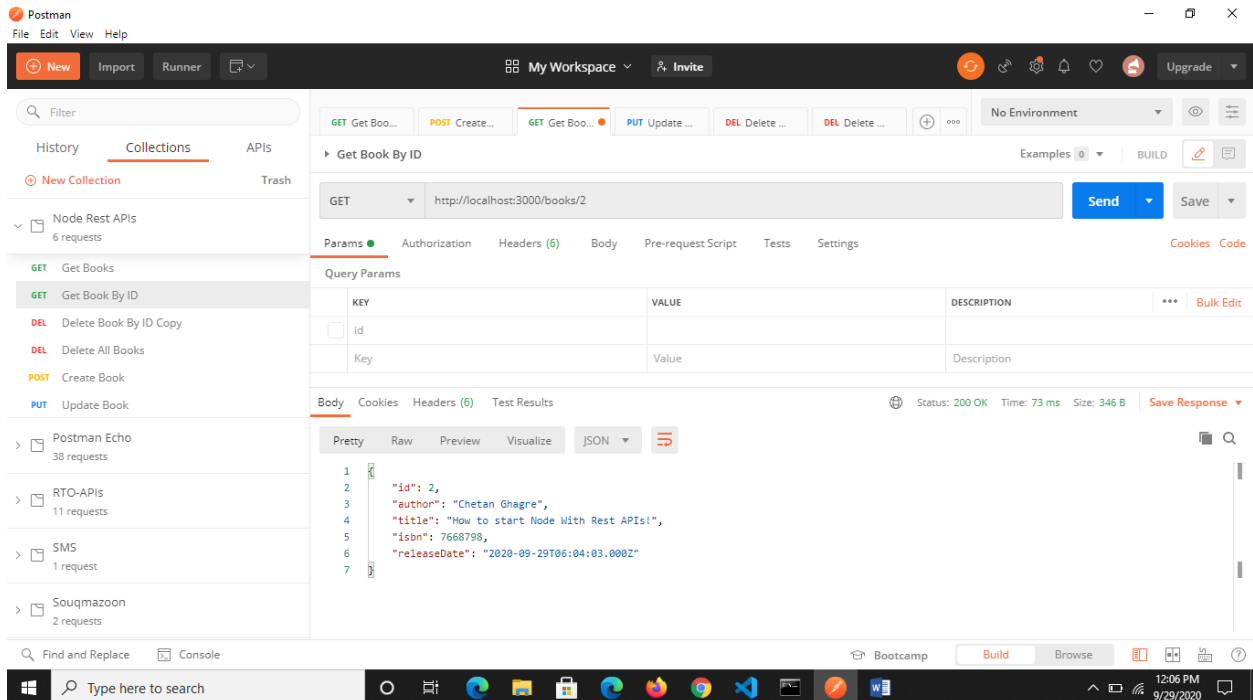
- **Create a new Book using POST /book Api**



- **Retrieve all Books using GET /books Api**



- **Retrieve a single Book by id using GET /books/:bookId Api**



• Update a Book using PUT /books/:bookId Api

The screenshot shows the Postman application interface. On the left, a sidebar lists collections, with 'Node Rest APIs' expanded to show a list of requests including 'PUT Update Book'. The main workspace displays the details of the 'PUT Update Book' request. The URL is 'http://localhost:3000/books/2'. The 'Body' tab is selected, showing a JSON payload:

```
{  "author": "Chetan Ghagre",  "title": "How to start Node!",  "isbn": 7647657}
```

. The response status is '200 OK' with a time of '48 ms' and size of '291 B'. The response body is shown in the 'Body' tab as:

```
{  "id": "2",  "author": "Chetan Ghagre",  "title": "How to start Node!",  "isbn": 7647657}
```

• Delete a Book using DELETE /books/:bookId Api

The screenshot shows the Postman application interface. On the left, a sidebar lists collections, with 'Node Rest APIs' expanded to show a list of requests including 'DELETE Book By ID Copy'. The main workspace displays the details of the 'DELETE Book By ID Copy' request. The URL is 'http://localhost:3000/books/1'. The 'Body' tab is selected, showing a JSON payload:

```
{  "message": "Book was deleted successfully!"}
```

. The response status is '200 OK' with a time of '14 ms' and size of '256 B'. The response body is shown in the 'Body' tab as:

```
{  "message": "Book was deleted successfully!"}
```

- **Delete all Books using DELETE /books Api**

The screenshot shows the Postman application interface. On the left, the 'Collections' tab is active, displaying a list of API collections. The 'Delete All Books' collection is selected. The main panel shows a DELETE request to the endpoint `http://localhost:3000/books`. The request is configured with the following details:

- Method: DELETE
- URL: `http://localhost:3000/books`
- Headers (6):
- Body: (Empty)
- Pre-request Script: (Empty)
- Tests: (Empty)
- Settings: (Empty)

The response is displayed in the 'Body' tab, showing a JSON object with a success message:

```
{  "message": "All Books were deleted successfully!"}
```

The status of the response is 200 OK, with a time of 48 ms and a size of 262 B. The response is saved.