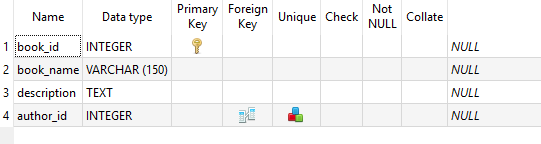
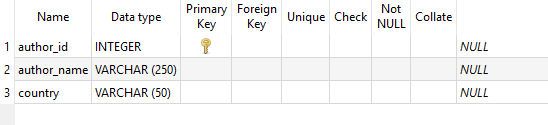
**REST API in Python**

**Database(sqlite3): book\_authors.db**

**tableName: Books**



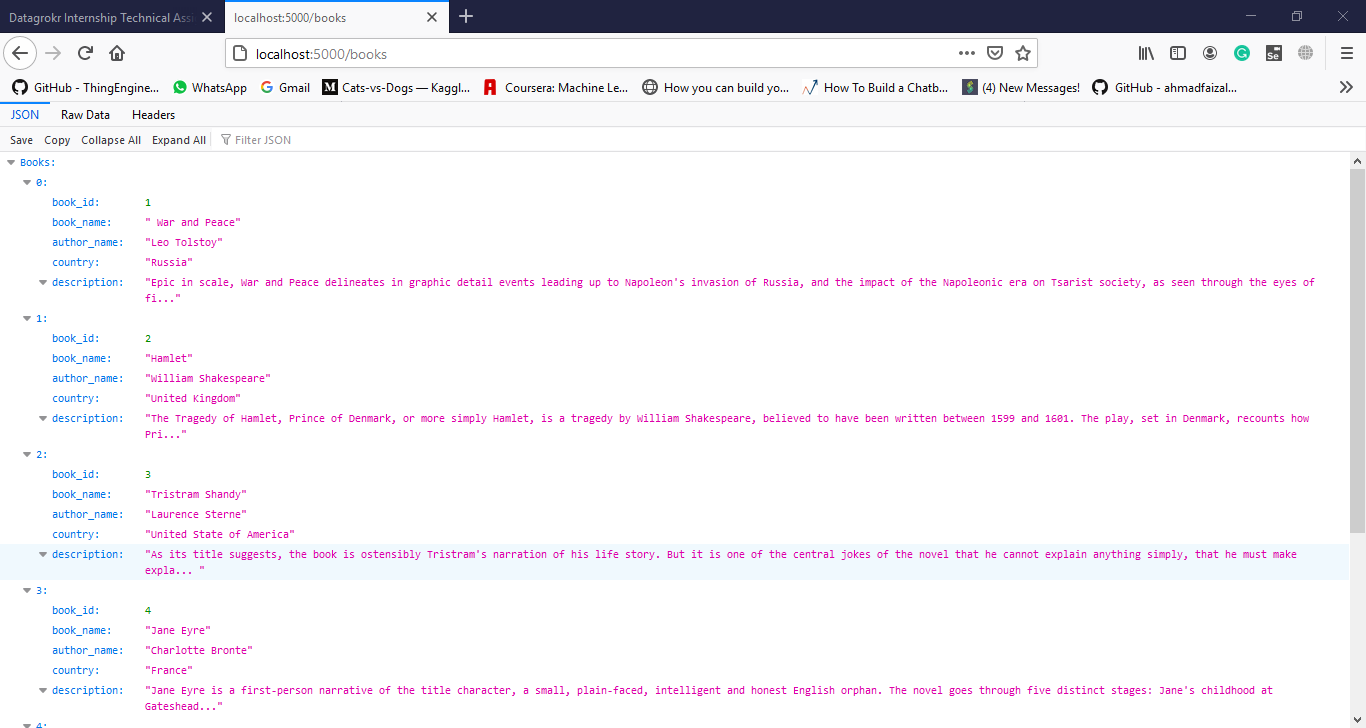
**tableName: Authors**



**Running Flask Script**

**$ python api.py**

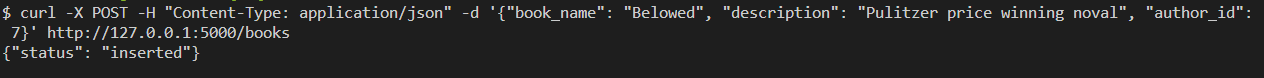
As you can see, the server has started on <http://127.0.0.1:5000>. Now, you can access the Flask server <http://127.0.0.1:5000>/books from web browser or API testing softwares such as CURL or Postman.



**CURL:**

**Adding Data Using POST in REST API (Insert)**

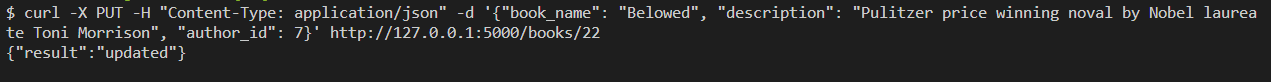
curl -X POST -H "Content-Type: application/json" -d '{"book\_name": "Belowed", "description": "Pulitzer price winning noval", "author\_id": 7}' http://127.0.0.1:5000/books



**NOTE:** The data should be inserted.

**Updating Data Using POST in REST API (Update)**

curl -X PUT -H "Content-Type: application/json" -d '{"book\_name": "Belowed", "description": "Pulitzer price winning noval by Nobel laureate Toni Morrison", "author\_id": 7}' http://127.0.0.1:5000/books/22

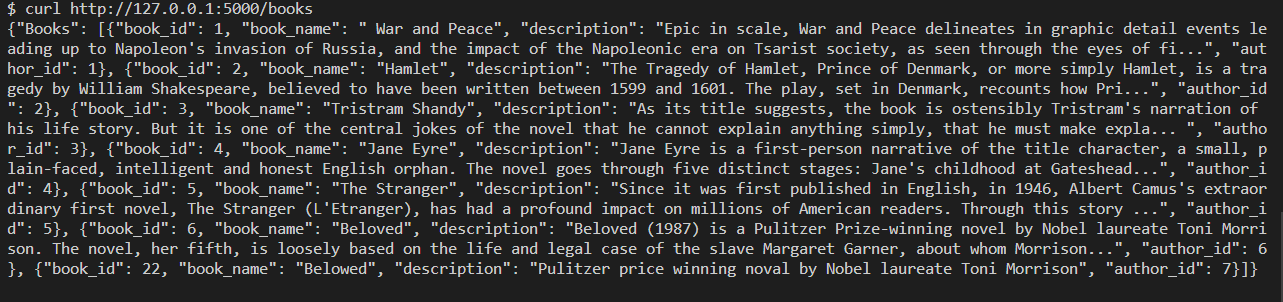


**NOTE:** The data should be updated.

**Accessing Data Using GET in REST API (Select)**

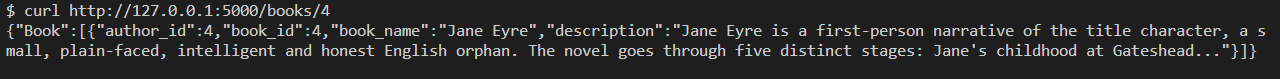
curl <http://127.0.0.1:5000/books>

As you can see, the accounts data is displayed as JSON format on GET request on /books endpoint.



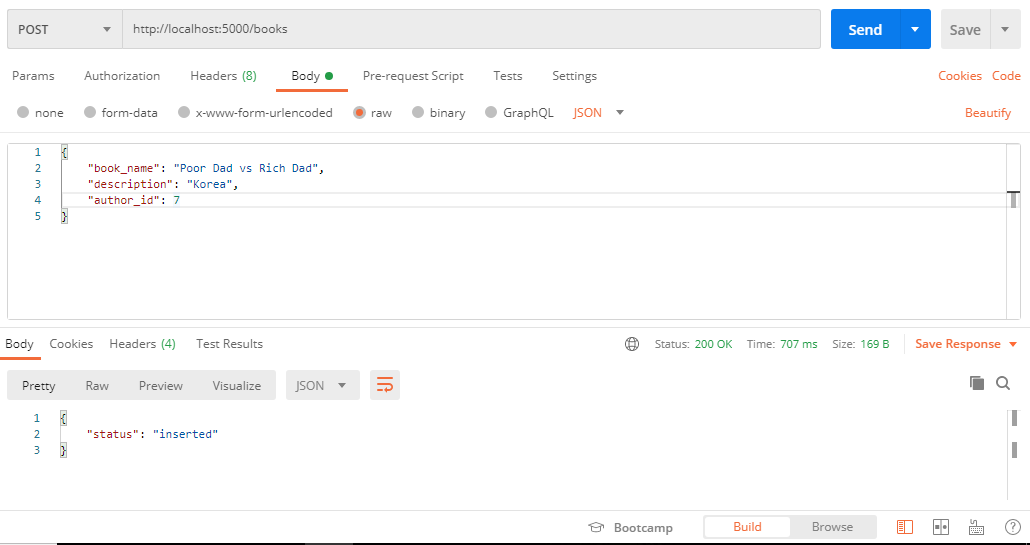
**Get specific book data**

curl http://127.0.0.1:5000/books/4

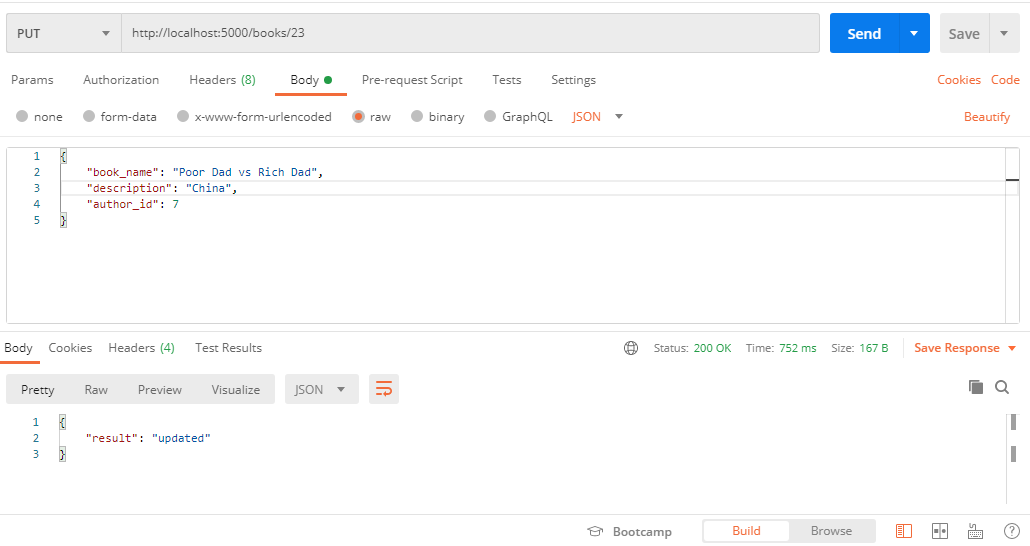


**Postman:**

**Adding Data Using POST in REST API (Insert)**



**Updating Data Using POST in REST API (Update)**



**Accessing Data Using GET in REST API (Select)**

