TESTING GUIDE FOR BOOKSTORE

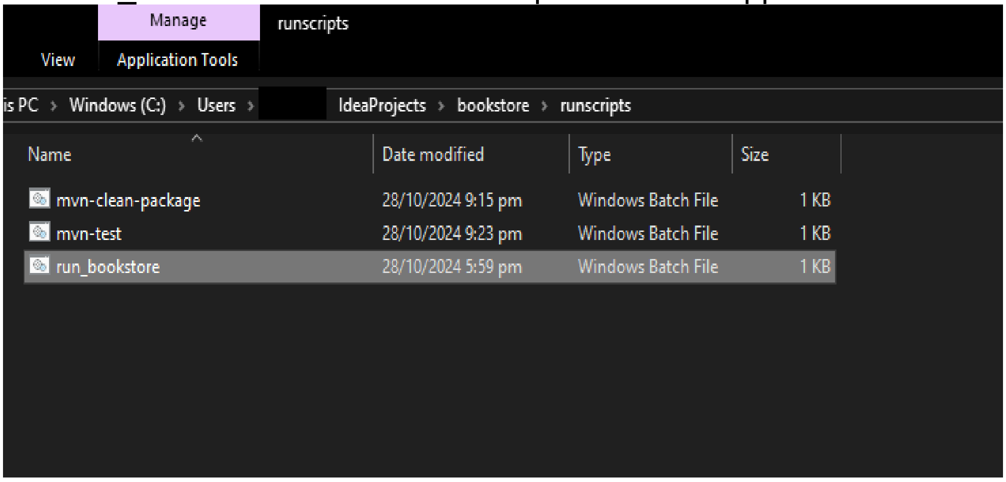
Prerequisites

Before starting with the Bookstore API project, ensure the following prerequisites are met:

1. **Java Development Kit (JDK) 17**:
   - Download and install JDK 17 or OpenJDK version that suits your system.
2. **Environment Variables Configuration**:
   - Set up the JAVA_HOME environment variable to point to your JDK installation directory.
   - Add the bin directory of the JDK to your system's PATH environment variable to run Java commands from the command line.
3. **Maven Project**:
   - Ensure that Bookstore Project have been built with the run scripts provided (Refer to README.MD)
   - Alternative install Apache Maven to use maven commands in command line to build the maven project.
4. **Git**:
   - Install Git to clone the repository from my Github
5. **IDE with Maven Support** (optional):
   - Consider using an Integrated Development Environment (IDE) such as IntelliJ IDEA or Eclipse with Maven support for easier project management and development.
6. **Postman or a Similar Tool**:
   - For testing the API endpoints, you can use Postman or any other API testing tool to send requests and receive responses from the Bookstore API.
   -
7. **Repository Access**:
   - Clone the repository for the Bookstore project from the GitHub repository to your local directory using the command:

Startup

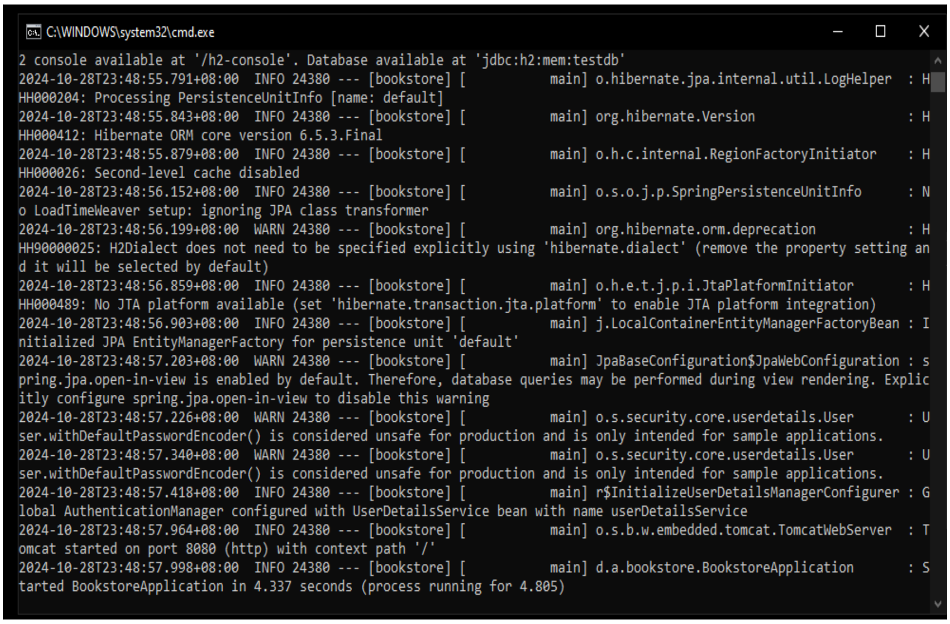
1. Launch Bookstore Application
   - Navigate to Project Root Folder then open /runscripts folder
   - Run run_bookstore.bat within /runscripts to launch Application



is PC › Windows (C:) › Users › [redacted] IdeaProjects › bookstore › runscripts

Name	Date modified	Type	Size
mvn-clean-package	28/10/2024 9:15 pm	Windows Batch File	1 KB
mvn-test	28/10/2024 9:23 pm	Windows Batch File	1 KB
run_bookstore	28/10/2024 5:59 pm	Windows Batch File	1 KB

2. Ensure the launch terminal window is opened
   - Embedded Jar file starts running in the background
   - Establish web application layer and the H2 Database
   - Keep the terminal window running during the duration of the test

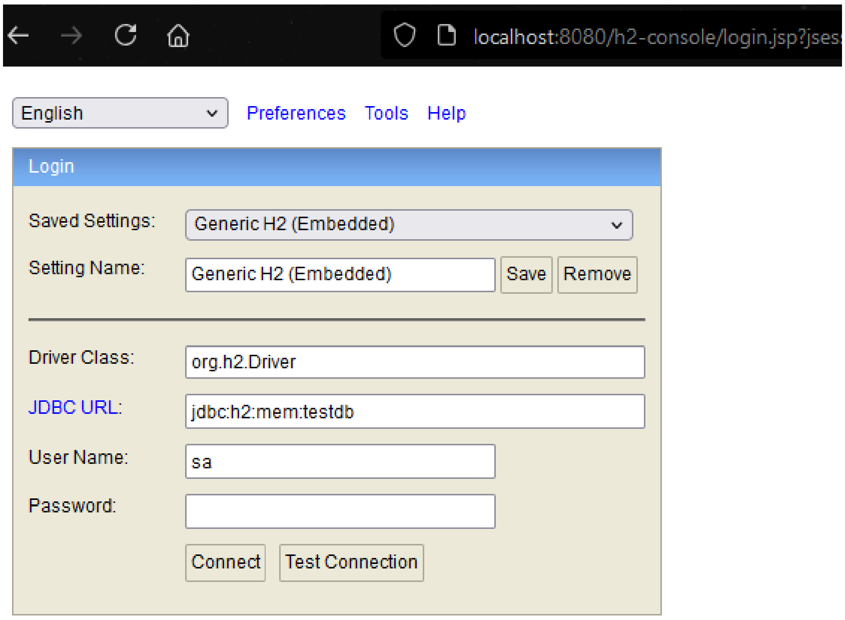


```
2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:testdb'
2024-10-28T23:48:55.791+08:00  INFO 24380 --- [bookstore] [           main] o.hibernate.jpa.internal.util.LogHelper  : H
HH000204: Processing PersistenceUnitInfo [name: default]
2024-10-28T23:48:55.843+08:00  INFO 24380 --- [bookstore] [           main] org.hibernate.Version                    : H
HH000412: Hibernate ORM core version 6.5.3.Final
2024-10-28T23:48:55.879+08:00  INFO 24380 --- [bookstore] [           main] o.h.c.internal.RegionFactoryInitiator    : H
HH000026: Second-level cache disabled
2024-10-28T23:48:56.152+08:00  INFO 24380 --- [bookstore] [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : N
o LoadTimeWeaver setup: ignoring JPA class transformer
2024-10-28T23:48:56.199+08:00  WARN 24380 --- [bookstore] [           main] org.hibernate.orm.deprecation            : H
HH90000025: H2Dialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting an
d it will be selected by default)
2024-10-28T23:48:56.859+08:00  INFO 24380 --- [bookstore] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : H
HH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-10-28T23:48:56.903+08:00  INFO 24380 --- [bookstore] [           main] j.LocalContainerEntityManagerFactoryBean : I
nitialized JPA EntityManagerFactory for persistence unit 'default'
2024-10-28T23:48:57.203+08:00  WARN 24380 --- [bookstore] [           main] JpaBaseConfiguration$JpaWebConfiguration : s
pring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explic
itly configure spring.jpa.open-in-view to disable this warning
2024-10-28T23:48:57.226+08:00  WARN 24380 --- [bookstore] [           main] o.s.security.core.userdetails.User       : U
ser.withDefaultPasswordEncoder() is considered unsafe for production and is only intended for sample applications.
2024-10-28T23:48:57.340+08:00  WARN 24380 --- [bookstore] [           main] o.s.security.core.userdetails.User       : U
ser.withDefaultPasswordEncoder() is considered unsafe for production and is only intended for sample applications.
2024-10-28T23:48:57.418+08:00  INFO 24380 --- [bookstore] [           main] r$InitializeUserDetailsManagerConfigurer : G
lobal AuthenticationManager configured with UserDetailsService bean with name userDetailsService
2024-10-28T23:48:57.964+08:00  INFO 24380 --- [bookstore] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : T
omcat started on port 8080 (http) with context path '/'
2024-10-28T23:48:57.998+08:00  INFO 24380 --- [bookstore] [           main] d.a.bookstore.BookstoreApplication       : S
tarted BookstoreApplication in 4.337 seconds (process running for 4.805)
```

3. Open Web Browser and enter URL for H2 Console for access
   - Key in “http://localhost:8080/h2-console/” in address bar and navigate
   - When H2 Console login screen appears, enter the following credentials
     - **Driver Class:** org.h2.Driver
     - **JDBC URL:** jdbc:h2:mem:testdb
     - **User Name:** sa

- **Password:** [blank]

- Click 'Test Connection' (Optional) to test
- Click 'Connect' to log in to H2 Console



localhost:8080/h2-console/login.jsp?jses

English

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded)

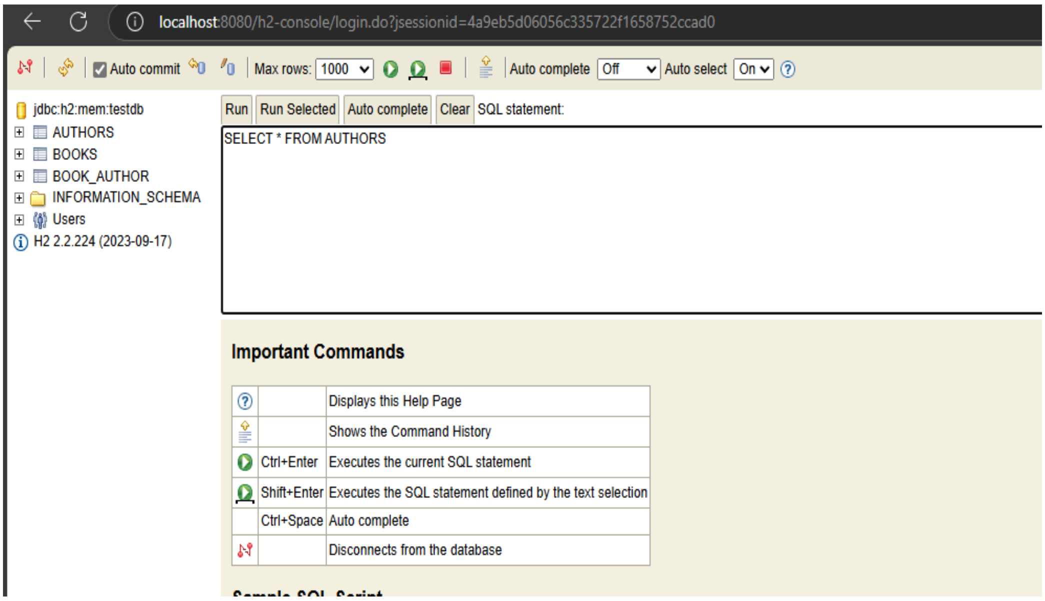
Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

4. H2 Console Screen
   - Ensure you are able to view and access the DB schema and its DB tables on the left side of the console
   - You can select any of the tables (AUTHORS, BOOKS, BOOKAUTHOR) on the left side to perform DB query for the particular table on the main frame window



localhost:8080/h2-console/login.do?jsessionid=4a9eb5d06056c335722f1658752ccad0

☑ Auto commit Max rows: 1000 Auto complete Off Auto select On

```
SELECT * FROM AUTHORS
```

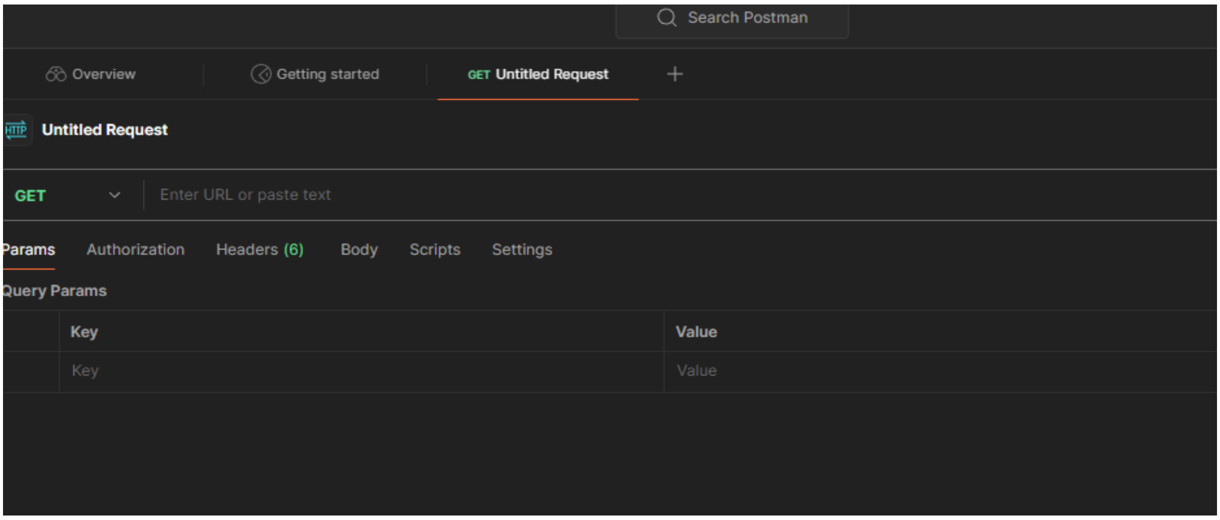
Important Commands

?		Displays this Help Page
		Shows the Command History
	Ctrl+Enter	Executes the current SQL statement
	Shift+Enter	Executes the SQL statement defined by the text selection
	Ctrl+Space	Auto complete
		Disconnects from the database

Sample SQL Script

5. Launch Postman application
   - Install if you have yet to do so
   - Will be used to test Bookstore API via RESTFUL calls

- After launching Postman, click on + icon beside “Getting Started” header to open new request window
- Request will be configured later during testing as part of the test steps

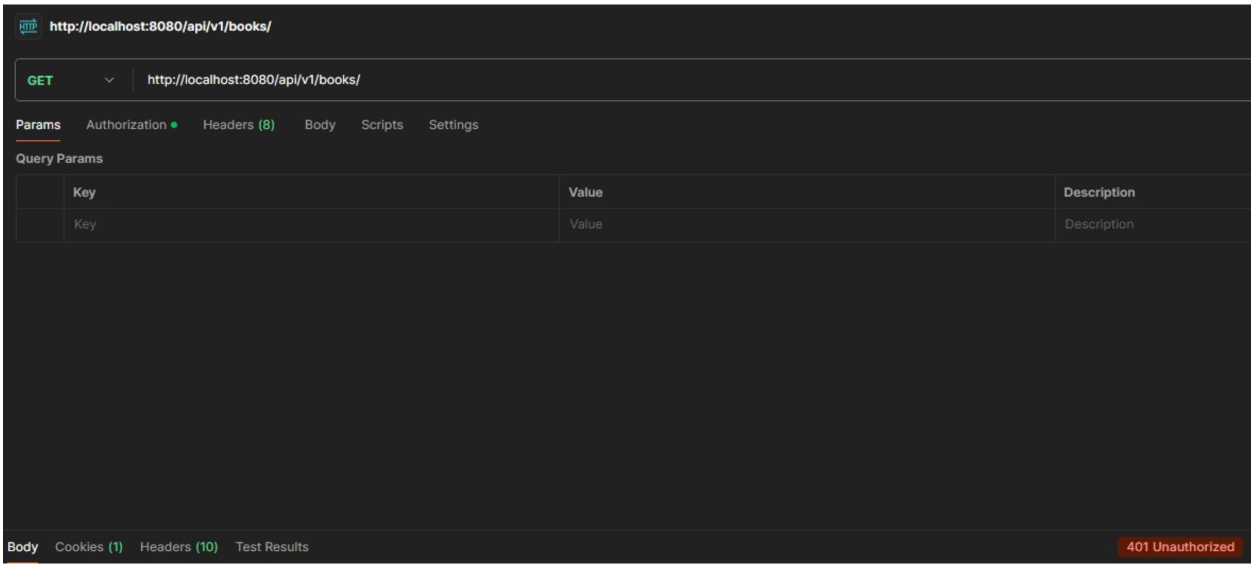


Testcases for User Role

Account for user roles has been configured with the following credentials for authentication and needs to be used in Postman Request later:

> **Username:** user
> **Password:** user

1. Perform HTTP GET books without Authorization details
   - In the Postman Request window, enter the following in URL bar
     - http://localhost:8080/api/v1/books/
   - Select GET from the dropdown on the left of URL bar
   - Click Send should result in 401 Unauthorized Error from response due to Web Security feature preventing unauthenticated requests from proceeding as shown below



http://localhost:8080/api/v1/books/

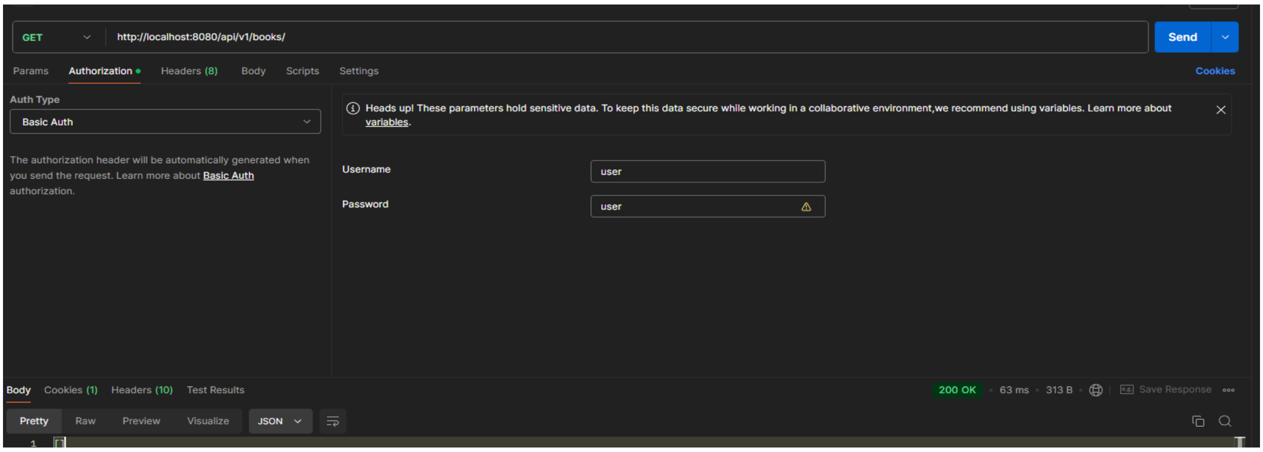
GET http://localhost:8080/api/v1/books/

Query Params

Key	Value	Description
Key	Value	Description

401 Unauthorized

2. Perform HTTP GET books with Authorization details of User Role
   - Repeat steps in 1. for URL bar vakue and GET selection
   - Click on the “Authorization” tab underneath URL bar
   - Select “Basic Auth” in the “Auth Type” drop down, then enter the following with the username password that was provided earlier
   - Click Send should return Status 200 OK which indicates successful response from Bookstore API
   - Json result will be as follow since no book or author record have been created



GET http://localhost:8080/api/v1/books/

Auth Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recommend using variables. Learn more about variables.

Username: user

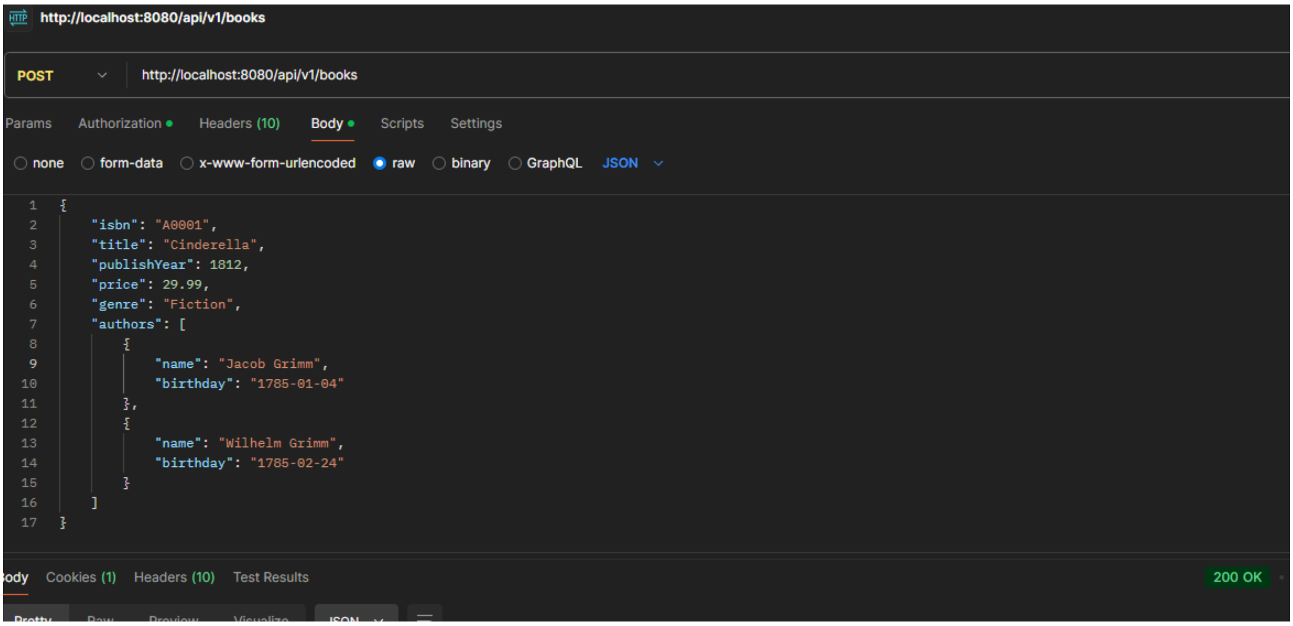
Password: user

200 OK · 63 ms · 313 B

3. Add Book with User Role
    - In the same Postman window or open a new window, enter the following in URL bar
        - http://localhost:8080/api/v1/books
    - Select POST from the dropdown on the left of URL bar
    - Select “Body” tab below URL bar, then select raw radio button
    - Ensure JSON format is selected and enter the following JSON message (Can refer to json files located in /src/test/data for samples)

```
{
   "isbn": "A0001",
   "title": "Cinderella",
   "publishYear": 1812,
   "price": 29.99,
   "genre": "Fiction",
   "authors": [
      {
         "name": "Jacob Grimm",
         "birthday": "1785-01-04"
      },
      {
         "name": "Wilhelm Grimm",
         "birthday": "1785-02-24"
      }
   ]
}
```

    - Click Send should return Status 200 OK with similar json content that was sent to bookstore API

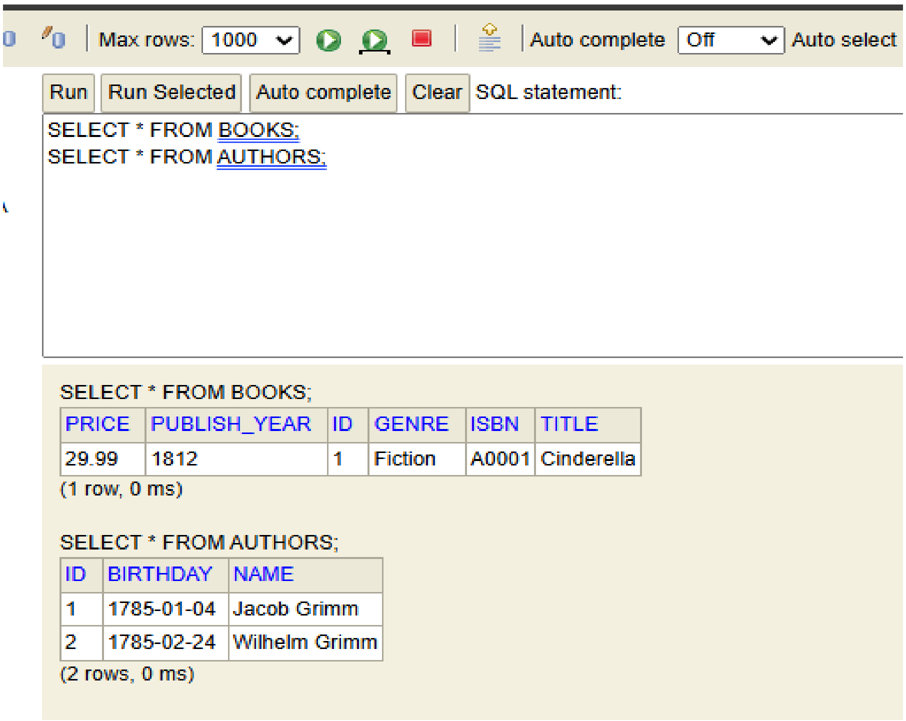


http://localhost:8080/api/v1/books
POST
http://localhost:8080/api/v1/books
Params
Authorization ●
Headers (10)
Body ●
Scripts
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
ody
Cookies (1)
Headers (10)
Test Results
200 OK

```
{
    "isbn": "A0001",
    "title": "Cinderella",
    "publishYear": 1812,
    "price": 29.99,
    "genre": "Fiction",
    "authors": [
        {
            "name": "Jacob Grimm",
            "birthday": "1785-01-04"
        },
        {
            "name": "Wilhelm Grimm",
            "birthday": "1785-02-24"
        }
    ]
}
```

    - Verify created records in H2 Console by performing SQL select queries for BOOKS and AUTHORS with the following

```
SELECT * FROM BOOKS;
SELECT * FROM AUTHORS;
```



```
SELECT * FROM BOOKS;
SELECT * FROM AUTHORS;
```

SELECT * FROM BOOKS;

PRICE	PUBLISH_YEAR	ID	GENRE	ISBN	TITLE
29.99	1812	1	Fiction	A0001	Cinderella

(1 row, 0 ms)

SELECT * FROM AUTHORS;

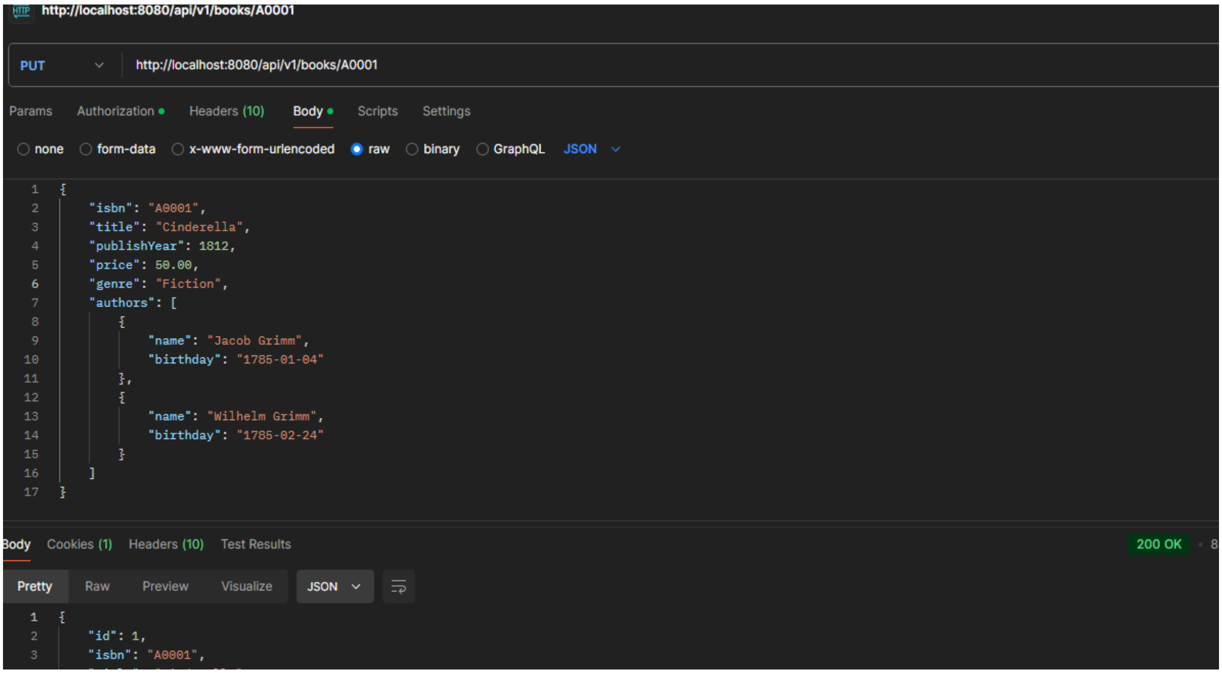
ID	BIRTHDAY	NAME
1	1785-01-04	Jacob Grimm
2	1785-02-24	Wilhelm Grimm

(2 rows, 0 ms)

4. Update Book with User Role
   - In the same Postman window or open a new window, enter the following in URL bar
     - http://localhost:8080/api/v1/books/{isbn}
   - Select PUT from the dropdown on the left of URL bar
   - Select “Body” tab below URL bar, then select raw radio button
   - Ensure JSON format is selected and enter the following JSON message (Notice Price has been changed)

```
{
   "isbn": "A0001",
   "title": "Cinderella",
   "publishYear": 1812,
   "price": 50.00,
   "genre": "Fiction",
   "authors": [
     {
       "name": "Jacob Grimm",
       "birthday": "1785-01-04"
     },
     {
       "name": "Wilhelm Grimm",
       "birthday": "1785-02-24"
     }
   ]
}
```

   - Click Send should return Status 200 OK with similar json content that was sent to bookstore API

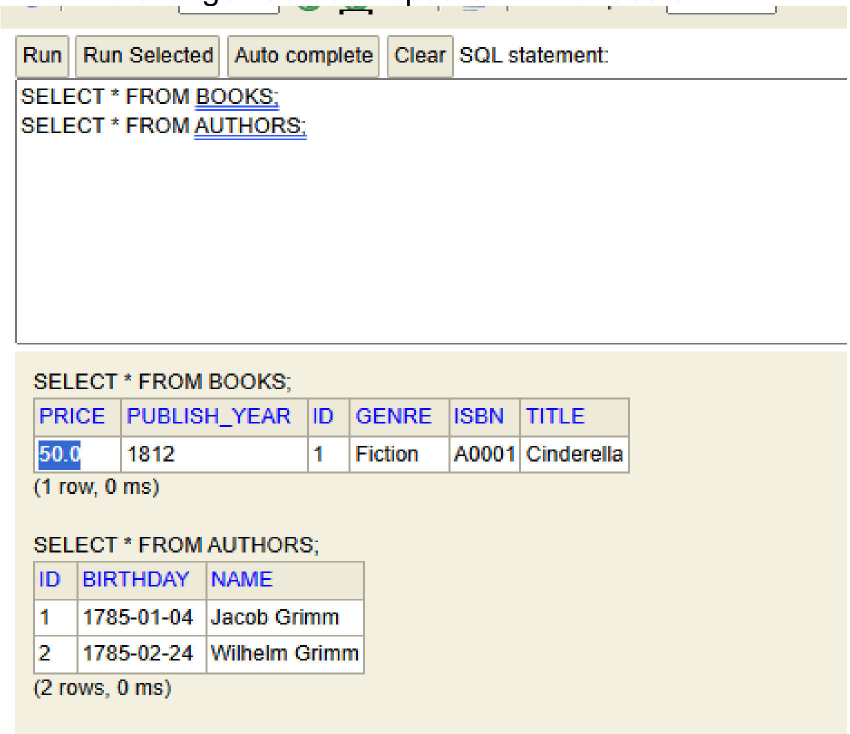


PUT http://localhost:8080/api/v1/books/A0001

```
{
    "isbn": "A0001",
    "title": "Cinderella",
    "publishYear": 1812,
    "price": 50.00,
    "genre": "Fiction",
    "authors": [
        {
            "name": "Jacob Grimm",
            "birthday": "1785-01-04"
        },
        {
            "name": "Wilhelm Grimm",
            "birthday": "1785-02-24"
        }
    ]
}
```

```
{
    "id": 1,
    "isbn": "A0001",
```

- Verify created records in H2 Console with simiar SQL queries as above, then check that the change has been updated in the record



```
SELECT * FROM BOOKS;
SELECT * FROM AUTHORS;
```

SELECT * FROM BOOKS;

PRICE	PUBLISH_YEAR	ID	GENRE	ISBN	TITLE
50.0	1812	1	Fiction	A0001	Cinderella

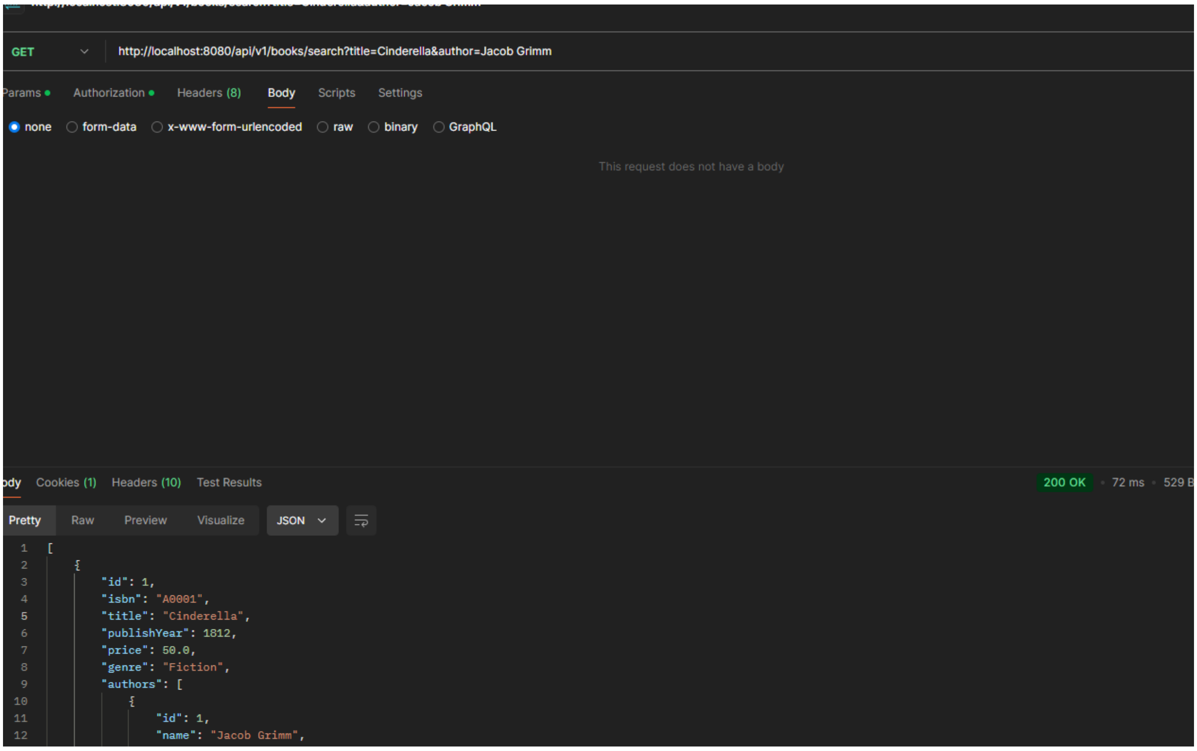
(1 row, 0 ms)

SELECT * FROM AUTHORS;

ID	BIRTHDAY	NAME
1	1785-01-04	Jacob Grimm
2	1785-02-24	Wilhelm Grimm

(2 rows, 0 ms)

5. Search Book with User Role
   - In the same Postman window or open a new window, enter any of the following in URL bar to search for book
     - http://localhost:8080/api/v1/books/search?title={title}&author={authorname}
     - http://localhost:8080/api/v1/books/search?title={title}
     - http://localhost:8080/api/v1/books/search?author={authorname}
   - Select GET from the dropdown on the left of URL bar
   - Select “Body” tab below URL bar, then select none radio button
   - Click Send should return Status 200 OK with json content of the book found



GET http://localhost:8080/api/v1/books/search?title=Cinderella&author=Jacob Grimm

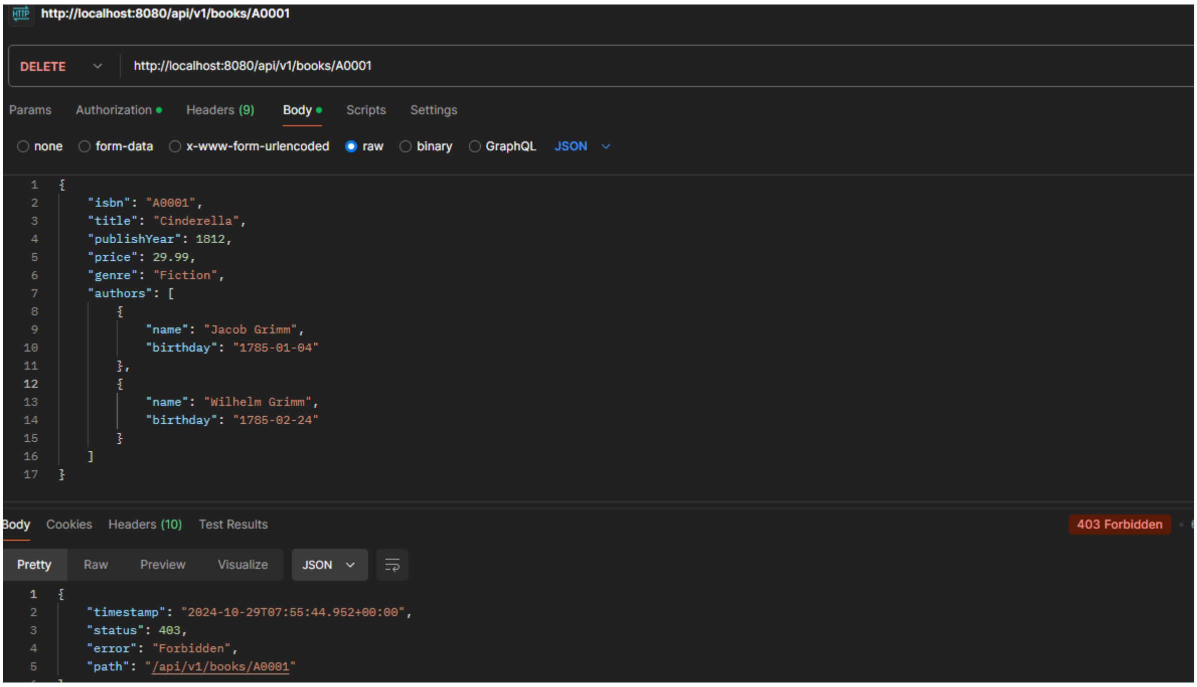
● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

200 OK · 72 ms · 529 B

```
[
    {
        "id": 1,
        "isbn": "A0001",
        "title": "Cinderella",
        "publishYear": 1812,
        "price": 50.0,
        "genre": "Fiction",
        "authors": [
            {
                "id": 1,
                "name": "Jacob Grimm",
```

6. Delete Book with User Role (Fail Attempt)
    - In the same Postman window or open a new window, enter the following in URL bar
        - http://localhost:8080/api/v1/books/{isbn}
    - Select DELETE from the dropdown on the left of URL bar
    - Select “Body” tab below URL bar, then select none radio button
    - Click Send should return Status 403 Forbidden with json content of the error as User is not allow to perform book deletion



http://localhost:8080/api/v1/books/A0001

DELETE http://localhost:8080/api/v1/books/A0001

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "isbn": "A0001",
    "title": "Cinderella",
    "publishYear": 1812,
    "price": 29.99,
    "genre": "Fiction",
    "authors": [
        {
            "name": "Jacob Grimm",
            "birthday": "1785-01-04"
        },
        {
            "name": "Wilhelm Grimm",
            "birthday": "1785-02-24"
        }
    ]
}
```

403 Forbidden

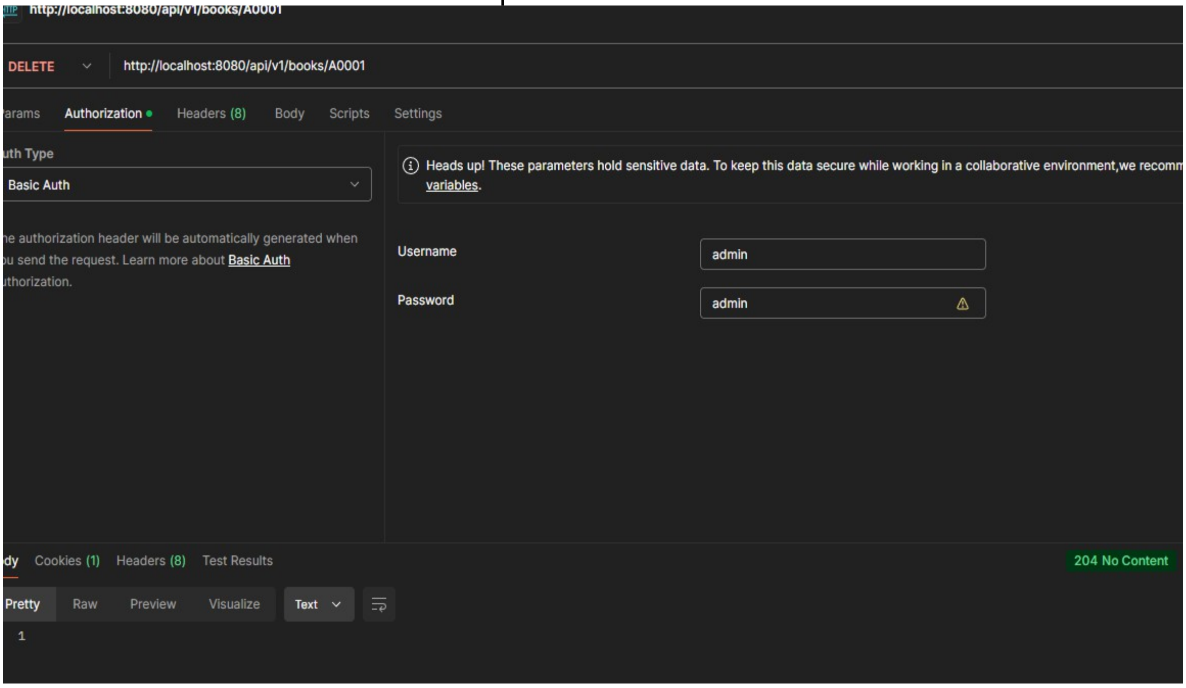
```
{
    "timestamp": "2024-10-29T07:55:44.952+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/v1/books/A0001"
```

7. Delete Book with Admin Role

- Replace the Basic Auth credential details in “Authorization” with that of Admin user account with the following

  Username: admin
  Password: admin

- Repeat steps in 6 to reattempt book deletion
- Click Send should return Status 204 No Content with no json content which indicates the book deletion operation is successful



http://localhost:8080/api/v1/books/A0001

DELETE http://localhost:8080/api/v1/books/A0001

uth Type: Basic Auth

he authorization header will be automatically generated when ou send the request. Learn more about Basic Auth uthorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recomm variables.

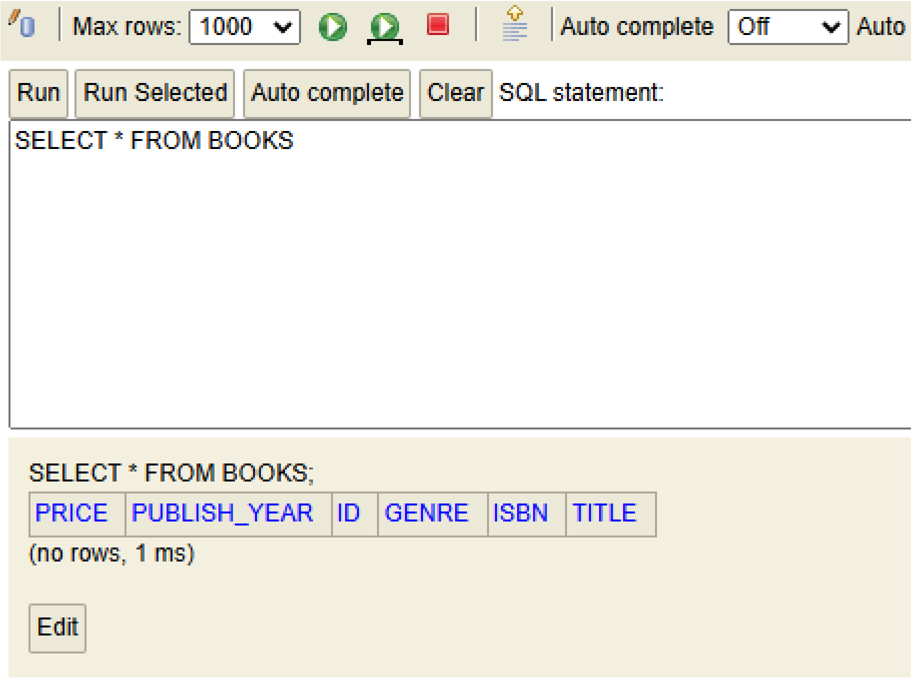
Username: admin

Password: admin

204 No Content

- Verify in H2 Console that the result for BOOKS has no records by performing the following SQL query

SELECT * FROM BOOKS



```
SELECT * FROM BOOKS
```

SELECT * FROM BOOKS;

PRICE	PUBLISH_YEAR	ID	GENRE	ISBN	TITLE

(no rows, 1 ms)

- Admin account can perform Add/Update actions on books and search for books as well, if you want to test it yourself. Results should be the same as user account test

8. Terminate the Application
    - End the application by closing the App Terminal Window