Bookstore assignment (Rough Docs)

1. **Installation:**
   1. PostgreSQL (import DB_backup file in pgAdmin).
   2. Import Maven project in eclipse.
   3. Change in the properties file in resource (hibernate database related info).
   4. Can be Run on tomcat server.

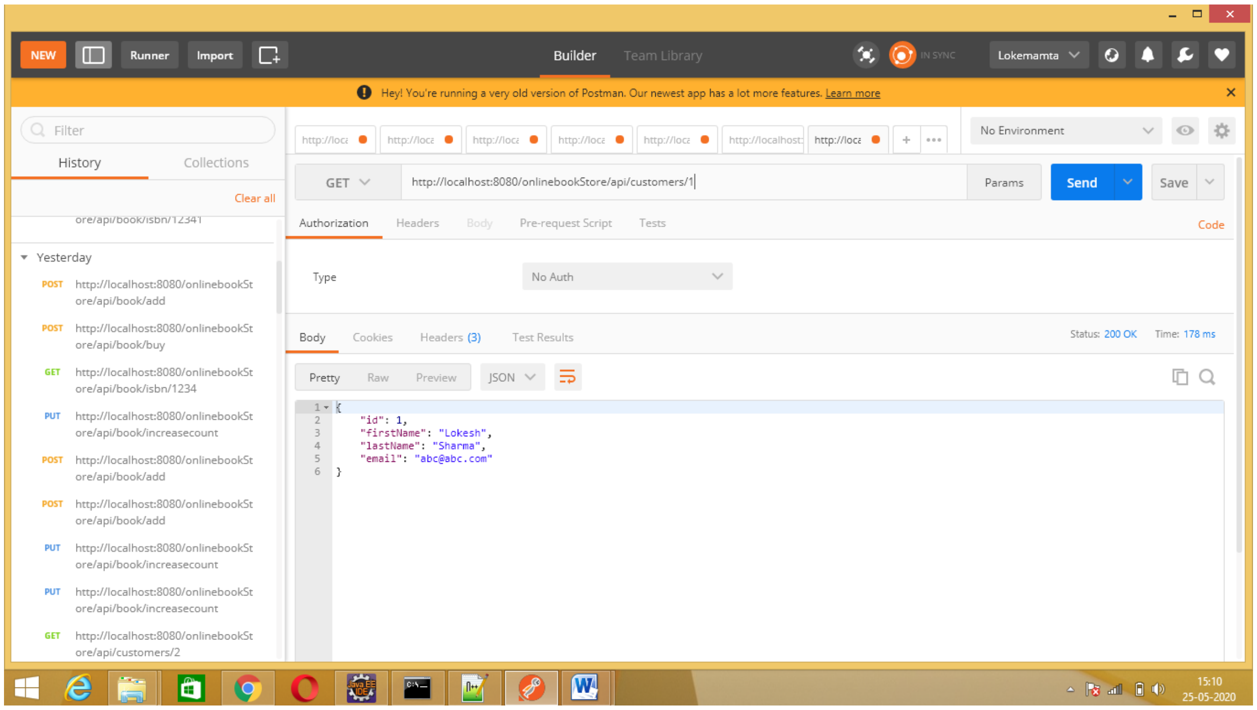
2. **Functionality Implemented:-**
   1. **Search book based on ISBN/Title/Author** :
      a. User can Search the book using ISBN/Title/Author. User can do partial search also using Title/Author.
   2. **Buy a book:**
      a. User can buy a book. (Two user can also buy the book together, " code can be used on multiple App servers as well").

      HQL Script : "**update Book as b set b.count = (b.count+:count) where b.count>0 and b.isbn= :isbn** "

      b. Created one more table(order) which will map customer and book.

      c. If book count ==0. It will send Notification to the staff. (till today it is checking the count value. Todo: send notification)
   3. **Add a Book to the store**
      a. Staff can add the book: A book can be added by staff(till today, anybody can add the book. Todo: implement staff)
   4. **Search media coverage about a book, given its ISBN:**
      a. Created API for media Coverage post

3. **APIS:**
   1. http://localhost:8080/onlinebookStore/api/customers/{customer_id} – get Request will require



ore/api/book/isbn/12341

Yesterday

POST http://localhost:8080/onlinebookStore/api/book/add
POST http://localhost:8080/onlinebookStore/api/book/buy
GET http://localhost:8080/onlinebookStore/api/book/isbn/1234
PUT http://localhost:8080/onlinebookStore/api/book/increasecount
POST http://localhost:8080/onlinebookStore/api/book/add
POST http://localhost:8080/onlinebookStore/api/book/add
PUT http://localhost:8080/onlinebookStore/api/book/increasecount
PUT http://localhost:8080/onlinebookStore/api/book/increasecount
GET http://localhost:8080/onlinebookStore/api/customers/2

GET http://localhost:8080/onlinebookStore/api/customers/1

Type: No Auth

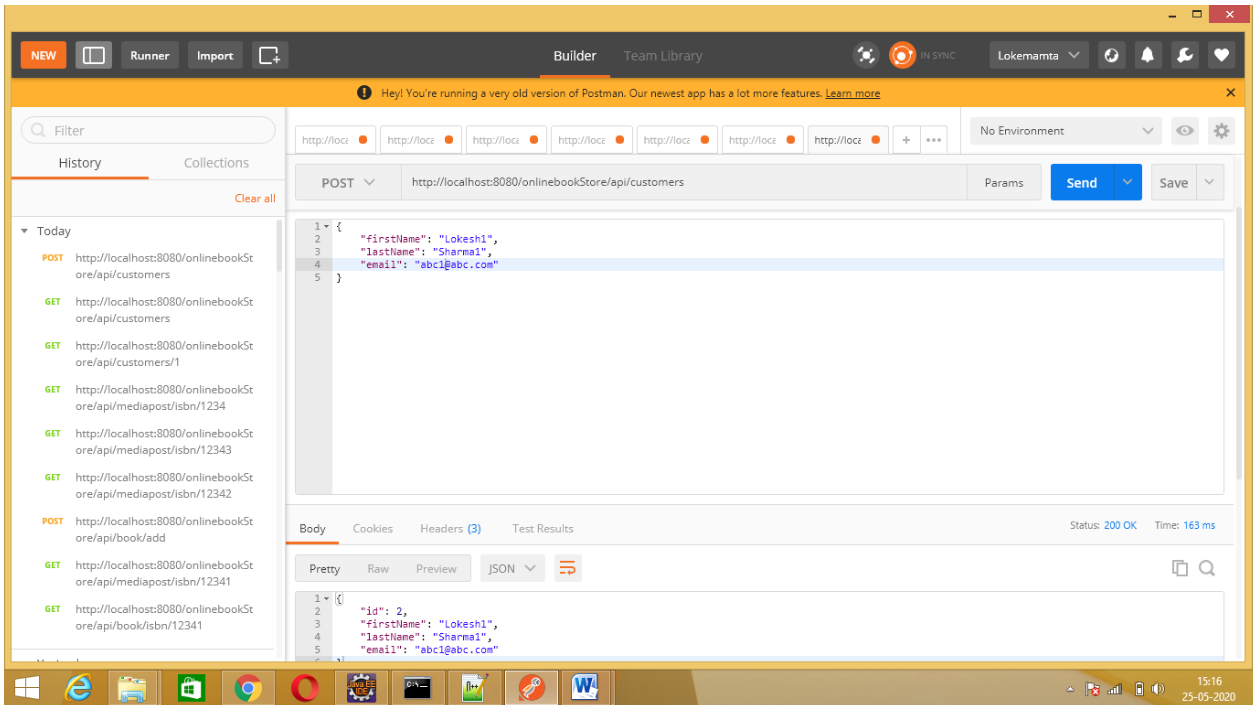
Status: 200 OK Time: 178 ms

```
{
    "id": 1,
    "firstName": "Lokesh",
    "lastName": "Sharma",
    "email": "abc@abc.com"
}
```

15:10
25-05-2020

2. http://localhost:8080/onlinebookStore/api/customers - post request to create customer JSON format –

```
{
   "firstName": "Lokesh",
   "lastName": "Sharma",
   "email": "abc@abc.com"
}
```



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

History | Collections | Clear all

Today

POST http://localhost:8080/onlinebookStore/api/customers
GET http://localhost:8080/onlinebookStore/api/customers
GET http://localhost:8080/onlinebookStore/api/customers/1
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/1234
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12343
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12342
POST http://localhost:8080/onlinebookStore/api/book/add
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12341
GET http://localhost:8080/onlinebookStore/api/book/isbn/12341

No Environment

POST http://localhost:8080/onlinebookStore/api/customers

```
{
    "firstName": "Lokesh1",
    "lastName": "Sharma1",
    "email": "abc1@abc.com"
}
```

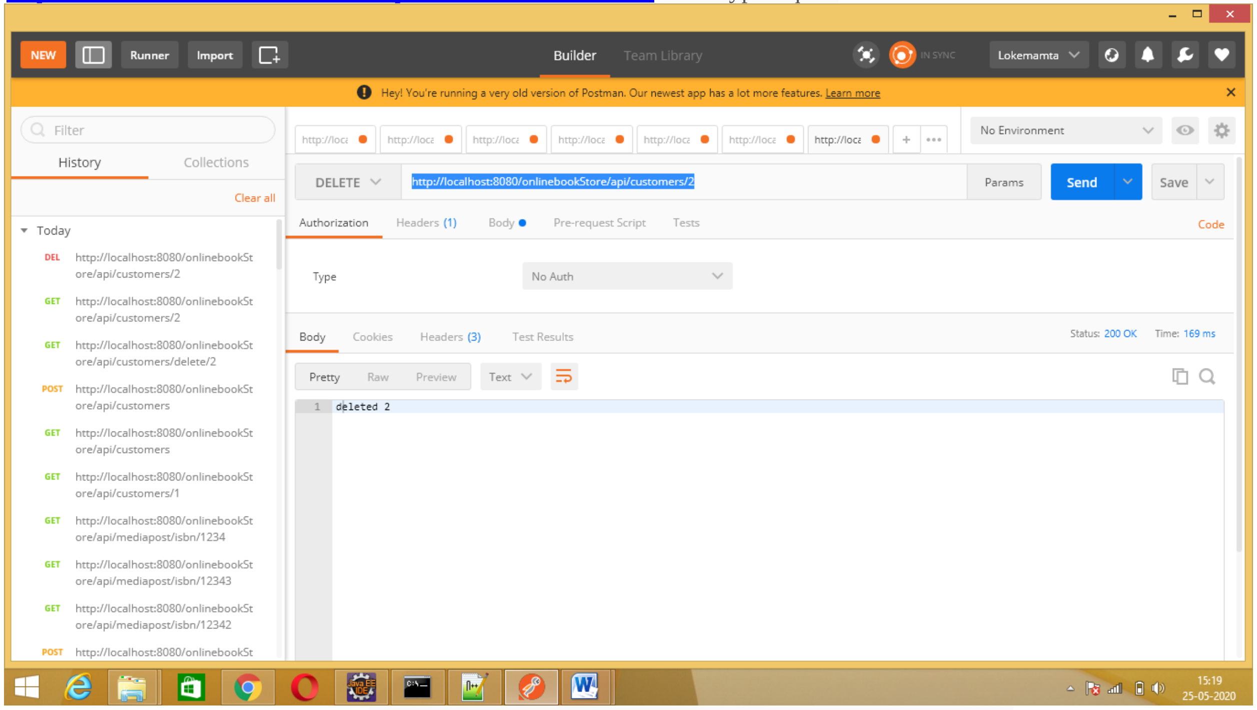
Body | Cookies | Headers (3) | Test Results — Status: 200 OK Time: 163 ms

Pretty | Raw | Preview | JSON

```
{
    "id": 2,
    "firstName": "Lokesh1",
    "lastName": "Sharma1",
    "email": "abc1@abc.com"
```

15:16
25-05-2020

3. http://localhost:8080/onlinebookStore/api/customers/{customerId} delete type request



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

History | Collections | Clear all

Today

DEL http://localhost:8080/onlinebookStore/api/customers/2
GET http://localhost:8080/onlinebookStore/api/customers/2
GET http://localhost:8080/onlinebookStore/api/customers/delete/2
POST http://localhost:8080/onlinebookStore/api/customers
GET http://localhost:8080/onlinebookStore/api/customers
GET http://localhost:8080/onlinebookStore/api/customers/1
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/1234
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12343
GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12342
POST http://localhost:8080/onlinebookSt

No Environment

DELETE http://localhost:8080/onlinebookStore/api/customers/2

Authorization | Headers (1) | Body | Pre-request Script | Tests | Code

Type: No Auth

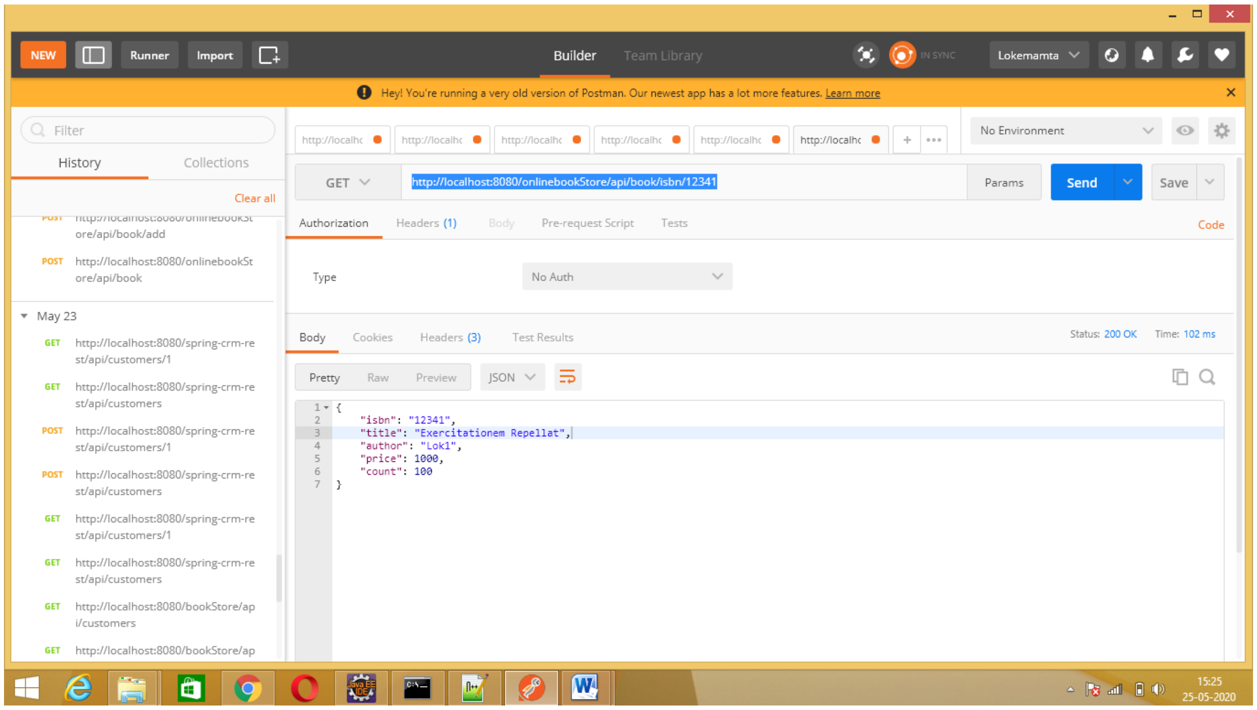
Body | Cookies | Headers (3) | Test Results — Status: 200 OK Time: 169 ms

Pretty | Raw | Preview | Text

```
deleted 2
```

15:19
25-05-2020

4. Get Book by ISBN (http://localhost:8080/onlinebookStore/api/book/isbn/{isbnNo} get Request**).**



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

History

- POST http://localhost:8080/onlinebookStore/api/book/add
- POST http://localhost:8080/onlinebookStore/api/book
- May 23
- GET http://localhost:8080/spring-crm-rest/api/customers/1
- GET http://localhost:8080/spring-crm-rest/api/customers
- POST http://localhost:8080/spring-crm-rest/api/customers/1
- POST http://localhost:8080/spring-crm-rest/api/customers
- GET http://localhost:8080/spring-crm-rest/api/customers/1
- GET http://localhost:8080/spring-crm-rest/api/customers
- GET http://localhost:8080/bookStore/api/customers
- GET http://localhost:8080/bookStore/ap

GET http://localhost:8080/onlinebookStore/api/book/isbn/12341

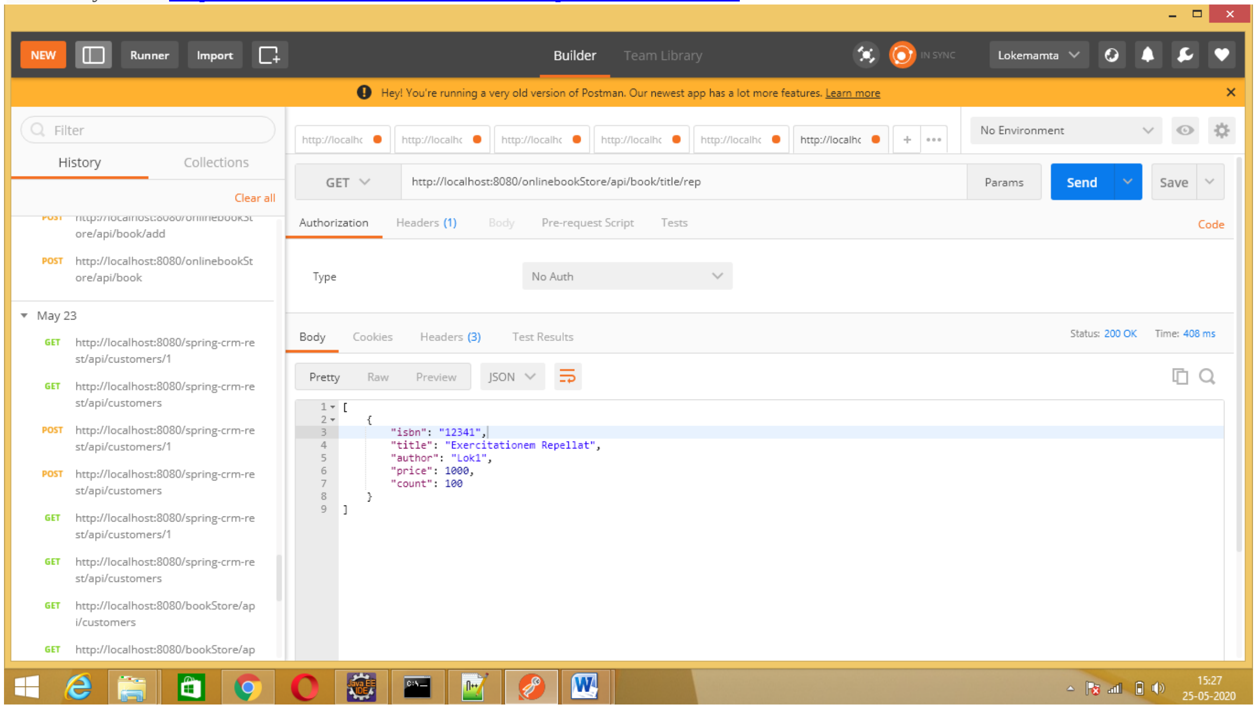
Authorization — Type: No Auth

Status: 200 OK Time: 102 ms

```
{
    "isbn": "12341",
    "title": "Exercitationem Repellat",
    "author": "Lok1",
    "price": 1000,
    "count": 100
}
```

15:25 25-05-2020

5. Search By title : http://localhost:8080/onlinebookStore/api/book/title/{title}



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

History

- POST http://localhost:8080/onlinebookStore/api/book/add
- POST http://localhost:8080/onlinebookStore/api/book
- May 23
- GET http://localhost:8080/spring-crm-rest/api/customers/1
- GET http://localhost:8080/spring-crm-rest/api/customers
- POST http://localhost:8080/spring-crm-rest/api/customers/1
- POST http://localhost:8080/spring-crm-rest/api/customers
- GET http://localhost:8080/spring-crm-rest/api/customers/1
- GET http://localhost:8080/spring-crm-rest/api/customers
- GET http://localhost:8080/bookStore/api/customers
- GET http://localhost:8080/bookStore/ap

GET http://localhost:8080/onlinebookStore/api/book/title/rep

Authorization — Type: No Auth

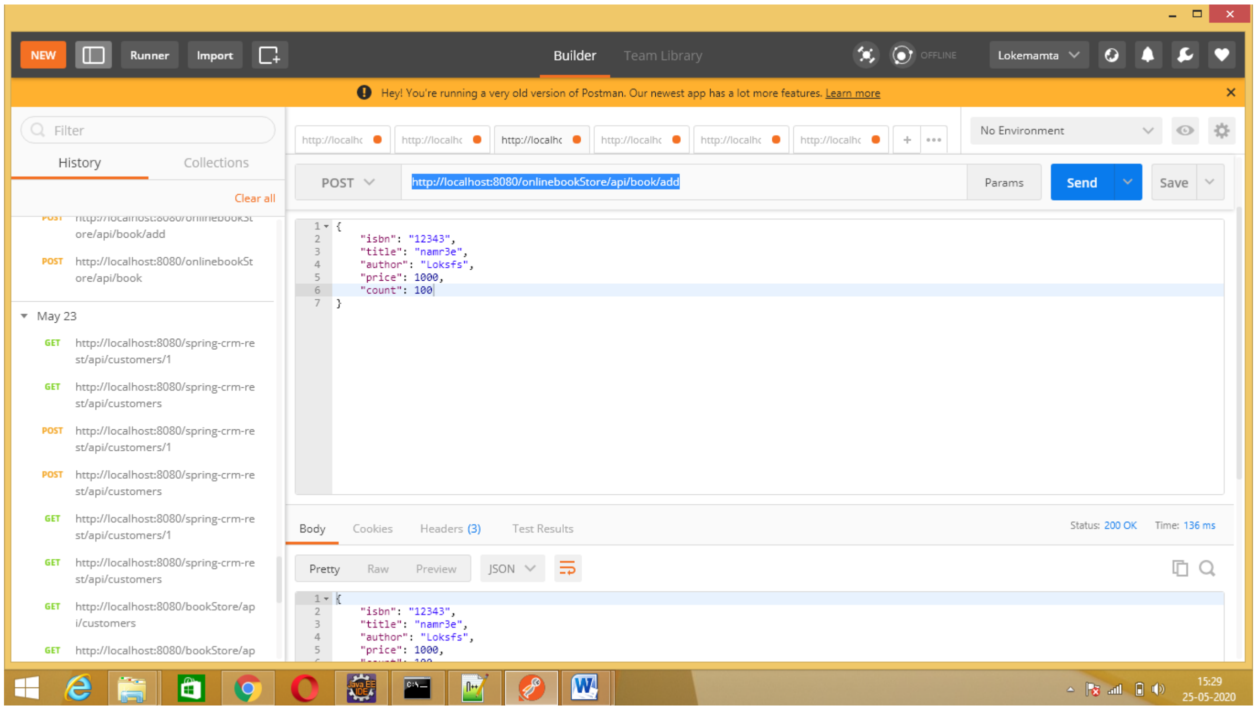
Status: 200 OK Time: 408 ms

```
[
    {
        "isbn": "12341",
        "title": "Exercitationem Repellat",
        "author": "Lok1",
        "price": 1000,
        "count": 100
    }
]
```

15:27 25-05-2020

6. Search By author : http://localhost:8080/onlinebookStore/api/book/author/{author}

7. Add book http://localhost:8080/onlinebookStore/api/book/add type -Post req



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

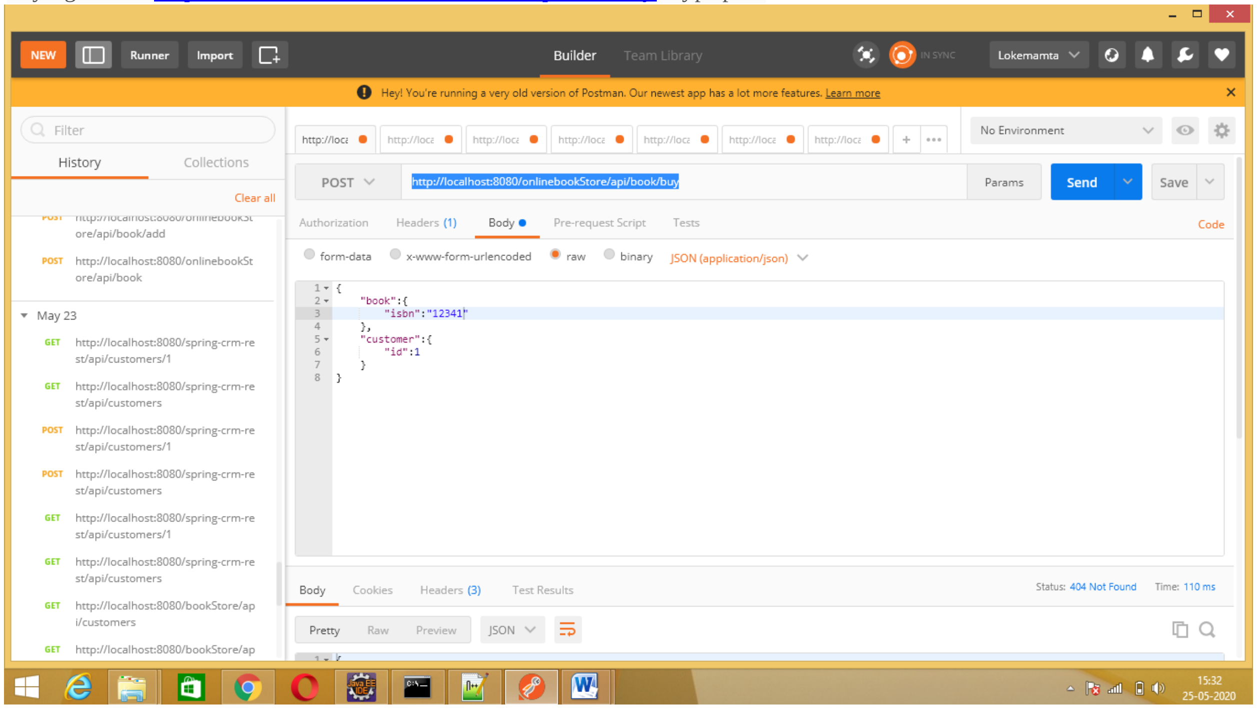
POST http://localhost:8080/onlinebookStore/api/book/add

```
{
    "isbn": "12343",
    "title": "namr3e",
    "author": "Loksfs",
    "price": 1000,
    "count": 100
}
```

Status: 200 OK Time: 136 ms

```
{
    "isbn": "12343",
    "title": "namr3e",
    "author": "Loksfs",
    "price": 1000,
```

8. Buying a book http://localhost:8080/onlinebookStore/api/book/buy - type post



Hey! You're running a very old version of Postman. Our newest app has a lot more features. Learn more

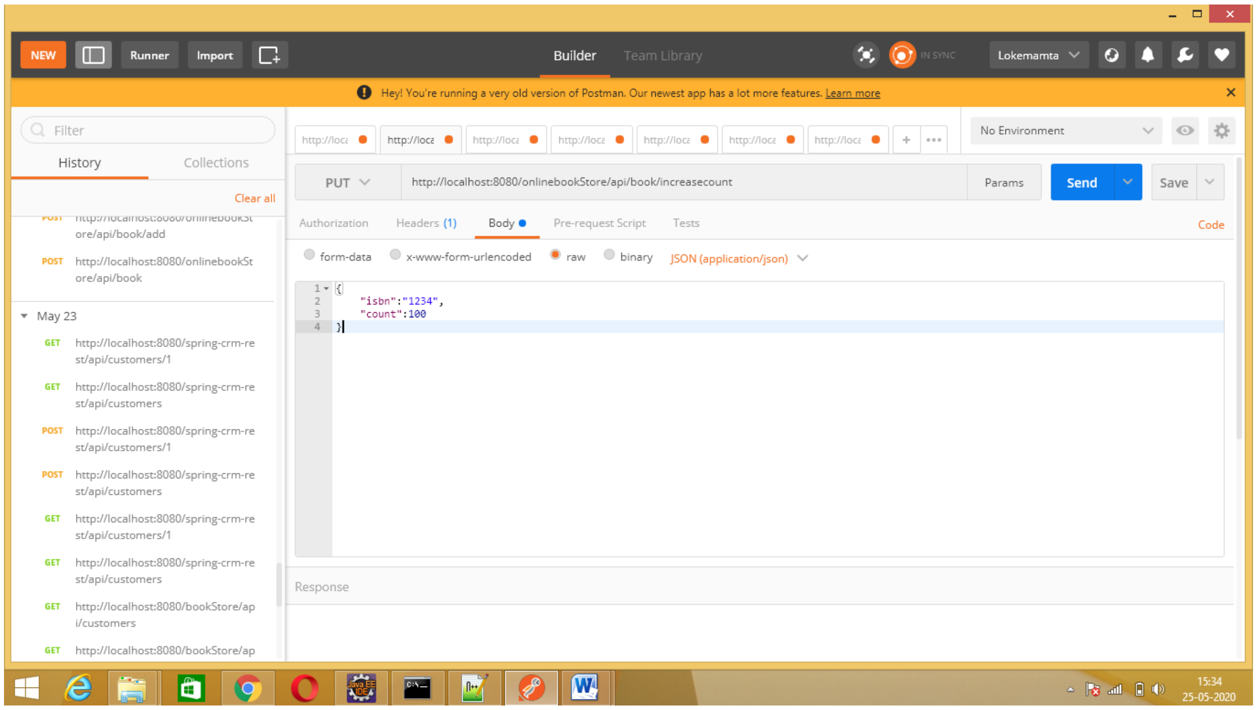
POST http://localhost:8080/onlinebookStore/api/book/buy

raw JSON (application/json)

```
{
    "book":{
        "isbn":"12341"
    },
    "customer":{
        "id":1
    }
}
```

Status: 404 Not Found Time: 110 ms

9. For increasing the count(avail) of the book in the shop - http://localhost:8080/onlinebookStore/api/book/increasecount type -put

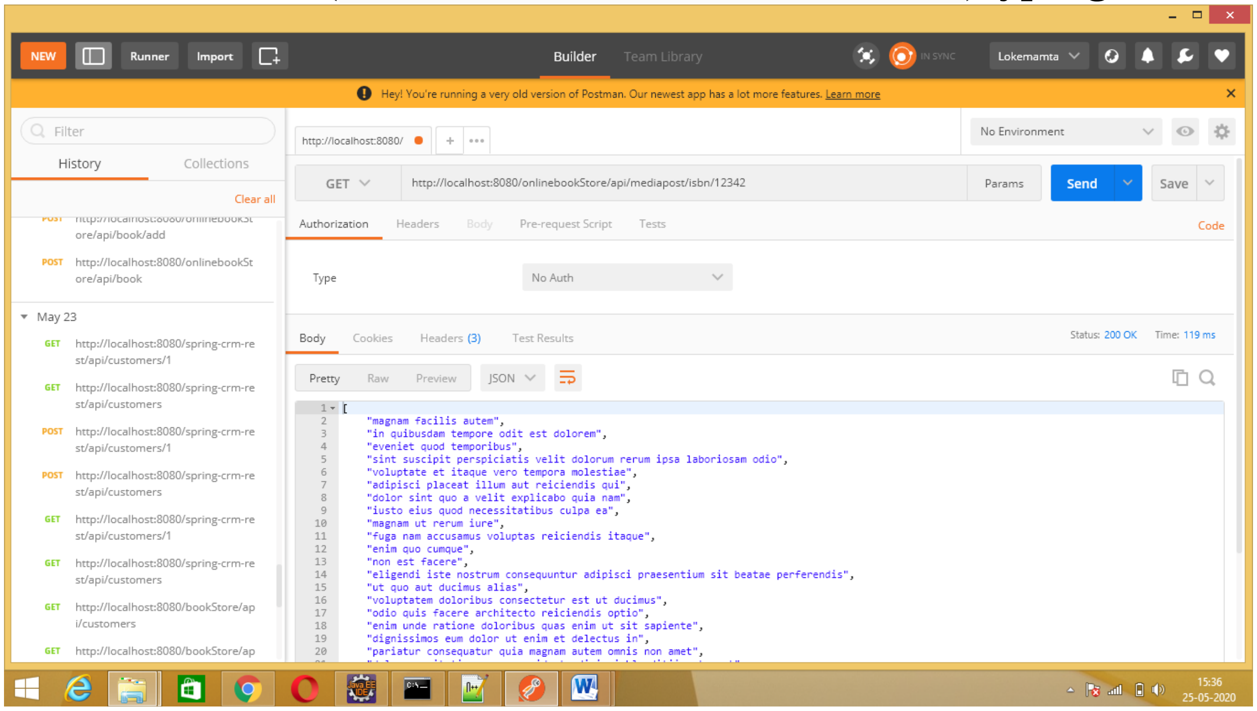


PUT http://localhost:8080/onlinebookStore/api/book/increasecount

Body: raw, JSON (application/json)

```
{
    "isbn":"1234",
    "count":100
}
```

10. Media Posts(http://localhost:8080/onlinebookStore/api/mediapost/isbn/12342) type -get



GET http://localhost:8080/onlinebookStore/api/mediapost/isbn/12342

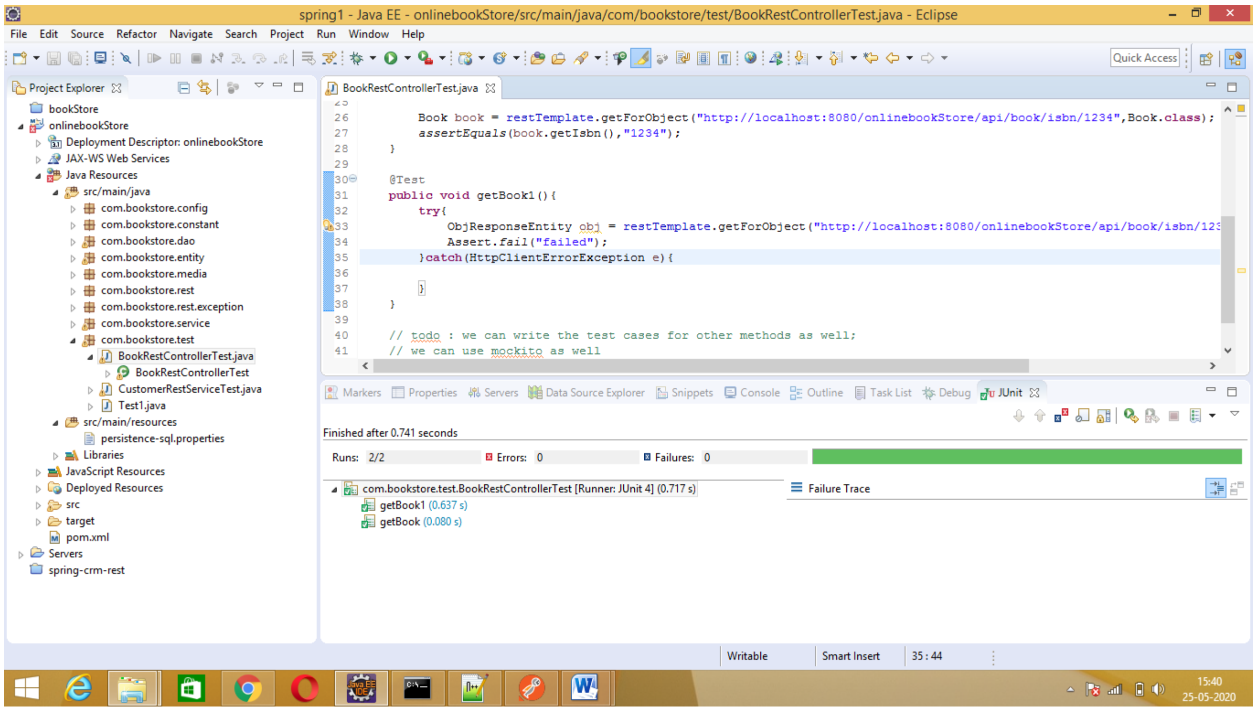
Type: No Auth

Status: 200 OK Time: 119 ms

```
[
    "magnam facilis autem",
    "in quibusdam tempore odit est dolorem",
    "eveniet quod temporibus",
    "sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio",
    "voluptate et itaque vero tempora molestiae",
    "adipisci placeat illum aut reiciendis qui",
    "dolor sint quo a velit explicabo quia nam",
    "iusto eius quod necessitatibus culpa ea",
    "magnam ut rerum iure",
    "fuga nam accusamus voluptas reiciendis itaque",
    "enim quo cumque",
    "non est facere",
    "eligendi iste nostrum consequuntur adipisci praesentium sit beatae perferendis",
    "ut quo aut ducimus alias",
    "voluptatem doloribus consectetur est ut ducimus",
    "odio quis facere architecto reiciendis optio",
    "enim unde ratione doloribus quas enim ut sit sapiente",
    "dignissimos eum dolor ut enim et delectus in",
    "pariatur consequatur quia magnam autem omnis non amet",
```

4. JUNIT Test Cases:

1.



spring1 - Java EE - onlinebookStore/src/main/java/com/bookstore/test/BookRestControllerTest.java - Eclipse

```
bookStore
onlinebookStore
  Deployment Descriptor: onlinebookStore
  JAX-WS Web Services
  Java Resources
    src/main/java
      com.bookstore.config
      com.bookstore.constant
      com.bookstore.dao
      com.bookstore.entity
      com.bookstore.media
      com.bookstore.rest
      com.bookstore.rest.exception
      com.bookstore.service
      com.bookstore.test
        BookRestControllerTest.java
          BookRestControllerTest
        CustomerRestServiceTest.java
        Test1.java
    src/main/resources
      persistence-sql.properties
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  pom.xml
Servers
spring-crm-rest
```

BookRestControllerTest.java

```java
        Book book = restTemplate.getForObject("http://localhost:8080/onlinebookStore/api/book/isbn/1234",Book.class);
        assertEquals(book.getIsbn(),"1234");
    }

    @Test
    public void getBook1(){
        try{
            ObjResponseEntity obj = restTemplate.getForObject("http://localhost:8080/onlinebookStore/api/book/isbn/123
            Assert.fail("failed");
        }catch(HttpClientErrorException e){

        }
    }

    // todo : we can write the test cases for other methods as well;
    // we can use mockito as well
```

```
Finished after 0.741 seconds

Runs: 2/2    Errors: 0    Failures: 0

com.bookstore.test.BookRestControllerTest [Runner: JUnit 4] (0.717 s)
    getBook1 (0.637 s)
    getBook (0.080 s)
```

5. **Docker container:**
    1. I have worked on CI (for example like git clone, git add ., git commit, git push, git merge,) part of the application. But I did not get a chance to work on cloud parts if I can get some time (like 4-5 days), I can learn and push it to git containers.
    2. Todo: Auth implementation, Person →staff and Customer inheritance, more better exception handling etc…