**Topic:** Bazar.com: A Multi-tier Online Book Store.

**Course**: Distributed and Operating Systems.

Name: 1- Lamya Hanani. 2- Amani Zaqzouq.

Catalog \_ Service:

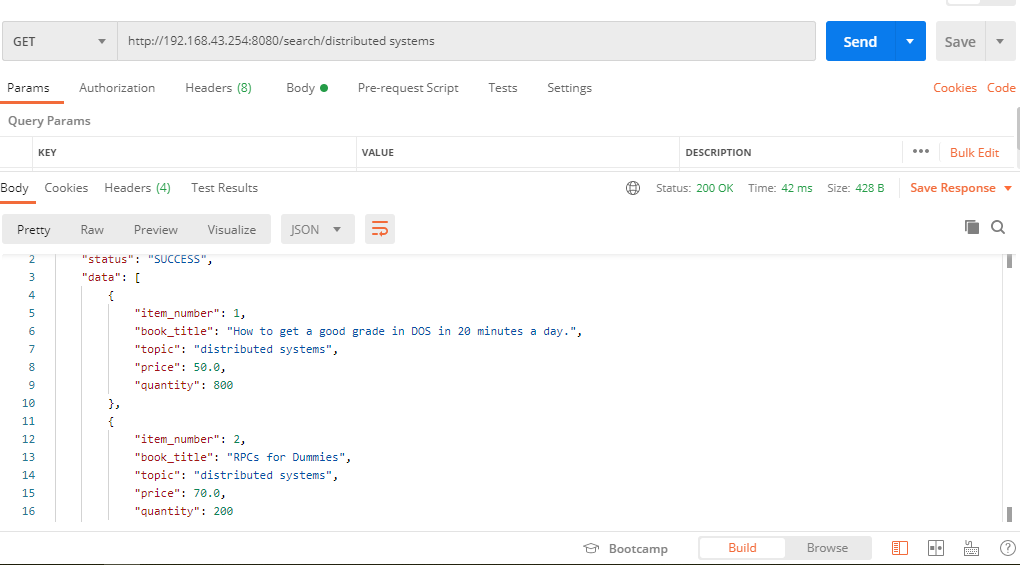
This service which depends in java spark library has 7 classes, three of them (User Exception, Status Response, Standard Response) for formatting messages that are send to users. The rest of them for:

1. Book class that identifies the book parameters (title, topic, price, count in stock, id) and implements the setters and getters for this parameters.
2. Catalog service interface : this identifies the catalog service operations : 1-getbooks(string topic) which returns all the books information that are belong to the determine topic(distributed systems or graduate school). 2-getbook (int id) which return the information for the specified book id. 3-Update(book b , string id) this function update the price or the quantity of the book. 4- isExist(int id) that return true if the book exist in the stock or the not.
3. Catalog\_service\_impl(): which implements all the functions in catalog\_service interface.
4. Spark\_main: this is the main for this service.

Catalog Service REST APIs :

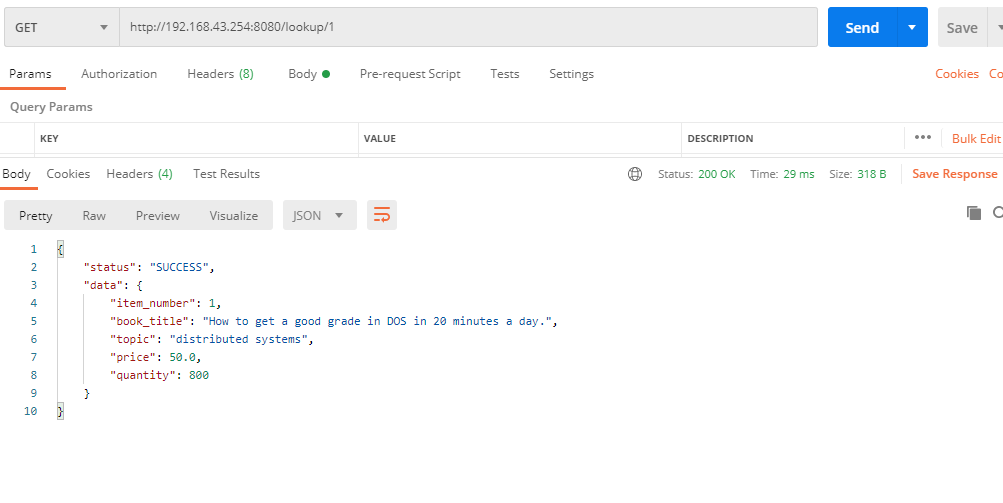
1. GET <http://192.168.43.254:8080/search/(topic)>.

e.x: <http://192.168.43.254:8080/search/distributed> systems:



1. GET <http://192.168.43.253:8080/lookup/(book> id)

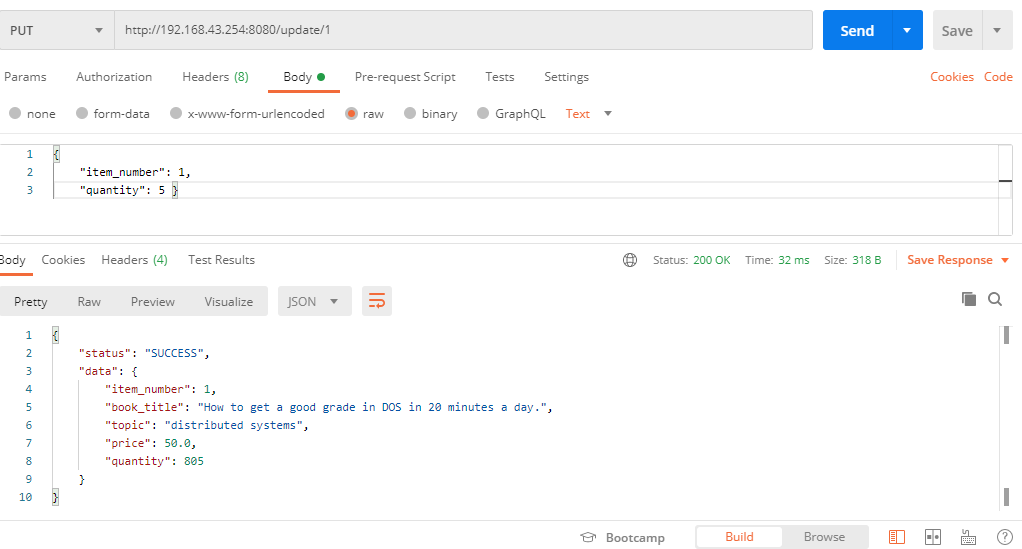
e.x: <http://192.168.43.253:8080/lookup/1>



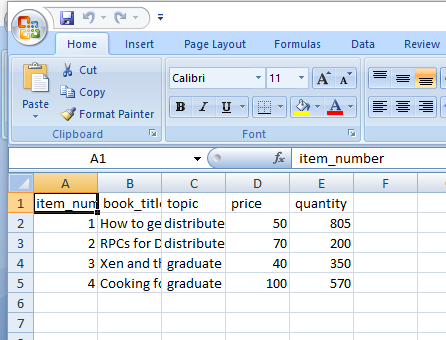
1. PUT <http://192.168.43.254:8080/update/(user> id) { price / quantity}

e.x: PUT <http://192.168.43.254:8080/update/1> {“item\_number”:1 , “quantity”:5}

This will increase book 1 quantity by 5, and the updated info for book 1 will be shown for the user to ensure that the quantity is modified in the CSV file.

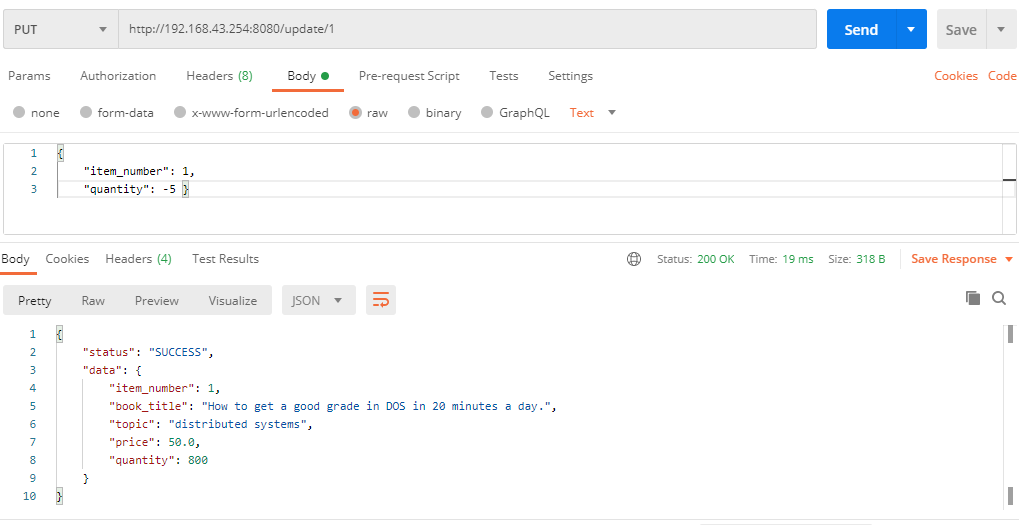


Updated CSV file:

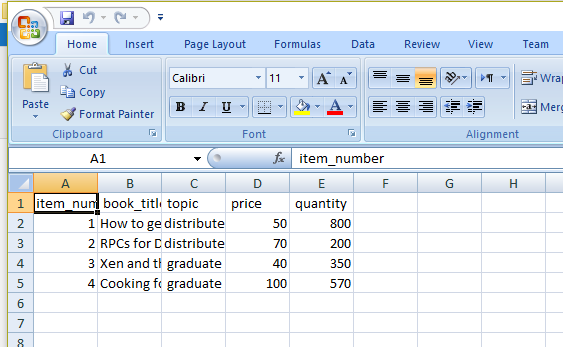


e.x2: PUT <http://192.168.43.254:8080/update/1> {“item\_number”:1 , “quantity”:-5}

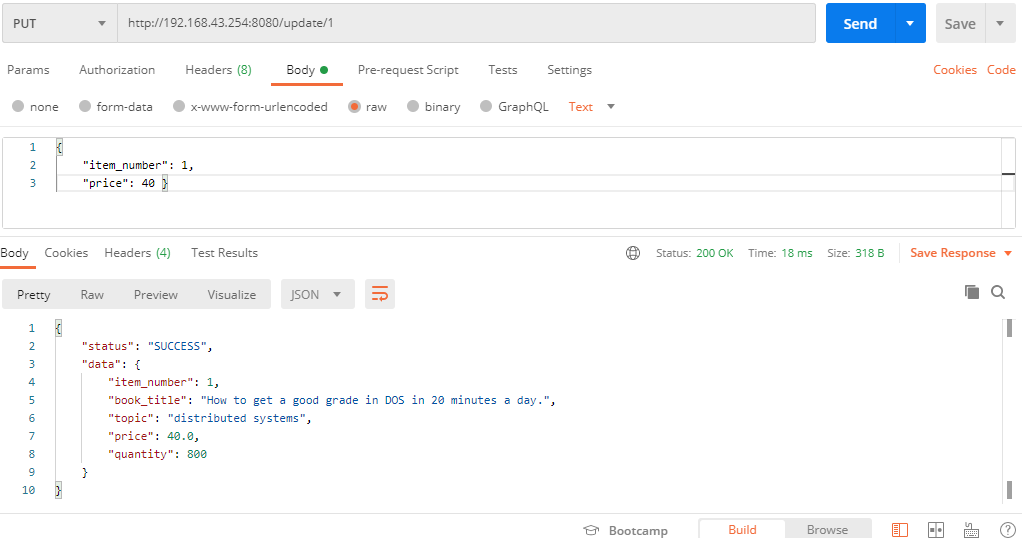
This will decrease the quantity by 5:



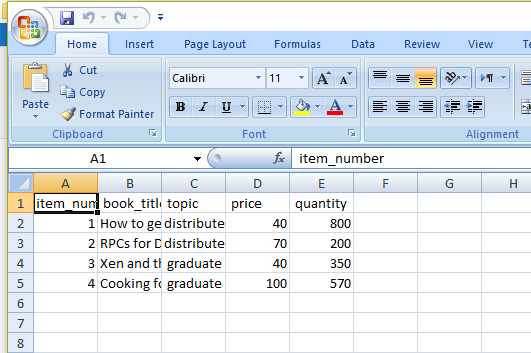
Updated CSV file:



e.x3: PUT <http://192.168.43.254:8080/update/1> {“item\_number”:1 , “price”: 40}

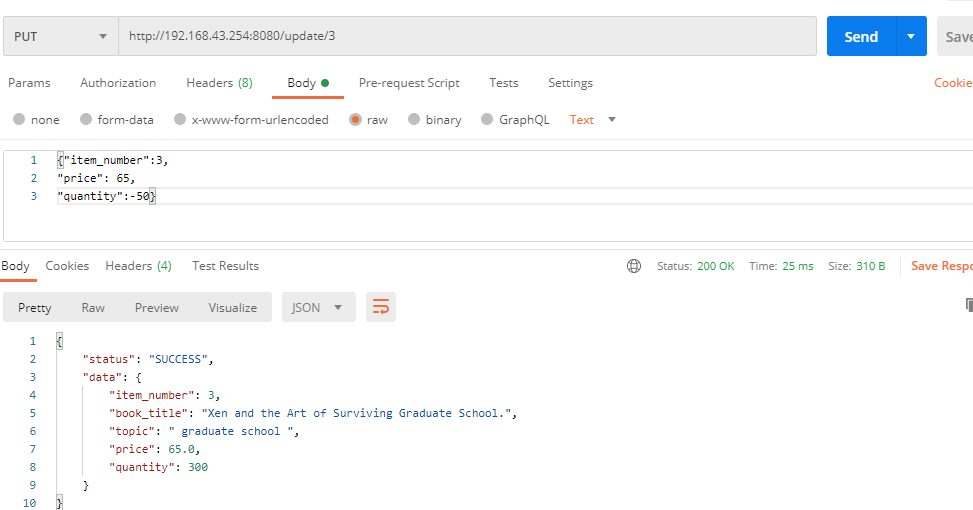


Updated CSV :

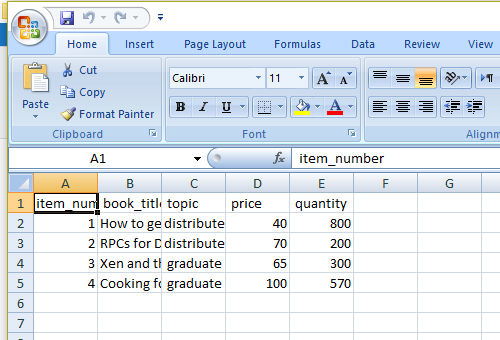


e.x4: You can update the price and the quantity at the same time:

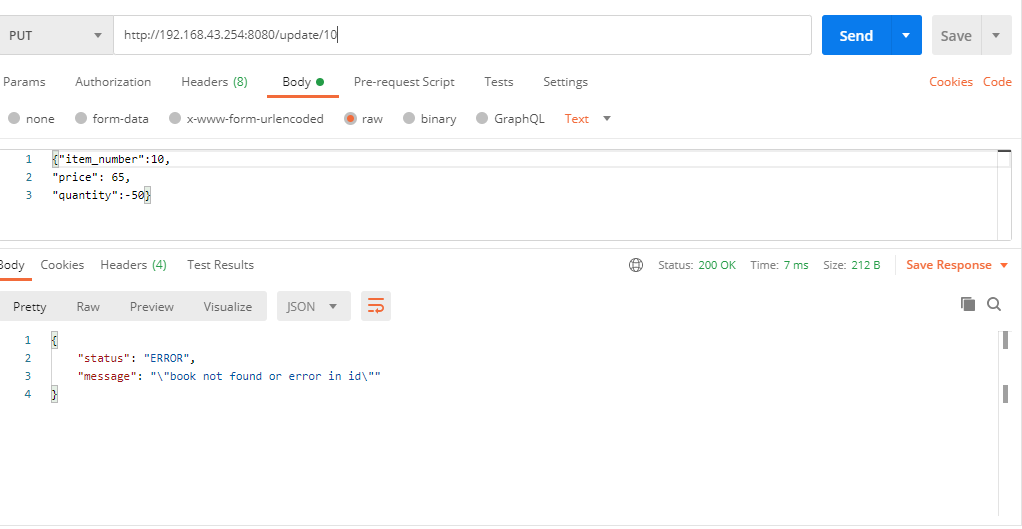
: PUT http://192.168.43.254:8080/update/3 {“item\_number”:3 , “price”: 60 , “quantity”: -50}

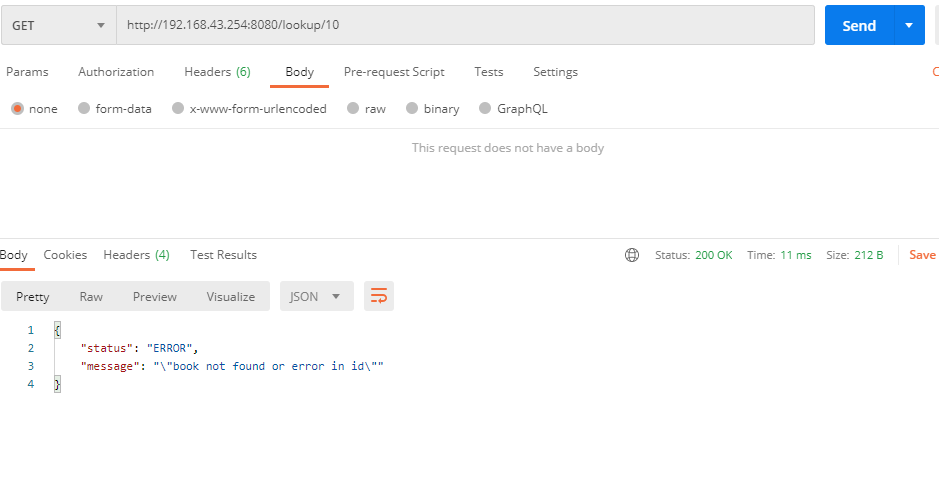


Updated CSV:

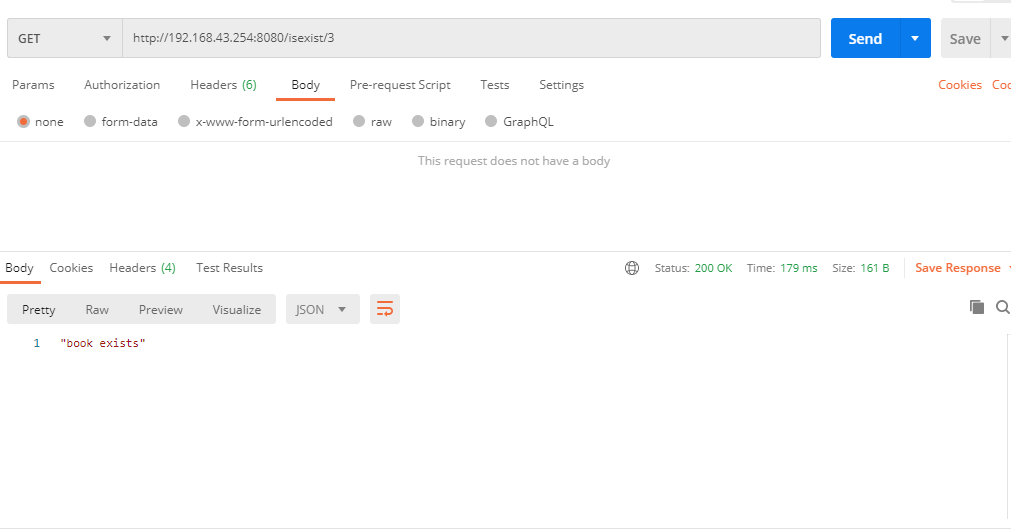


If the user tries to update or lookup a book that not exists in the Store, an error message will be shown for the user:





1. GET http:// [http://192.168.43.254:8080/isexist/ (user](http://192.168.43.254:8080/isexist/%20(user) id)
2. e.x1: GET http:// <http://192.168.43.254:8080/isexist/3> , it return “book exists” if it exists.



e.x2: GET http:// <http://192.168.43.254:8080/isexist/3>0 it return “book does not exist” if it does not exist in the store.

