CS 677 S21 Lab 3 - Bookstore

Test Cases Document

Submitted By

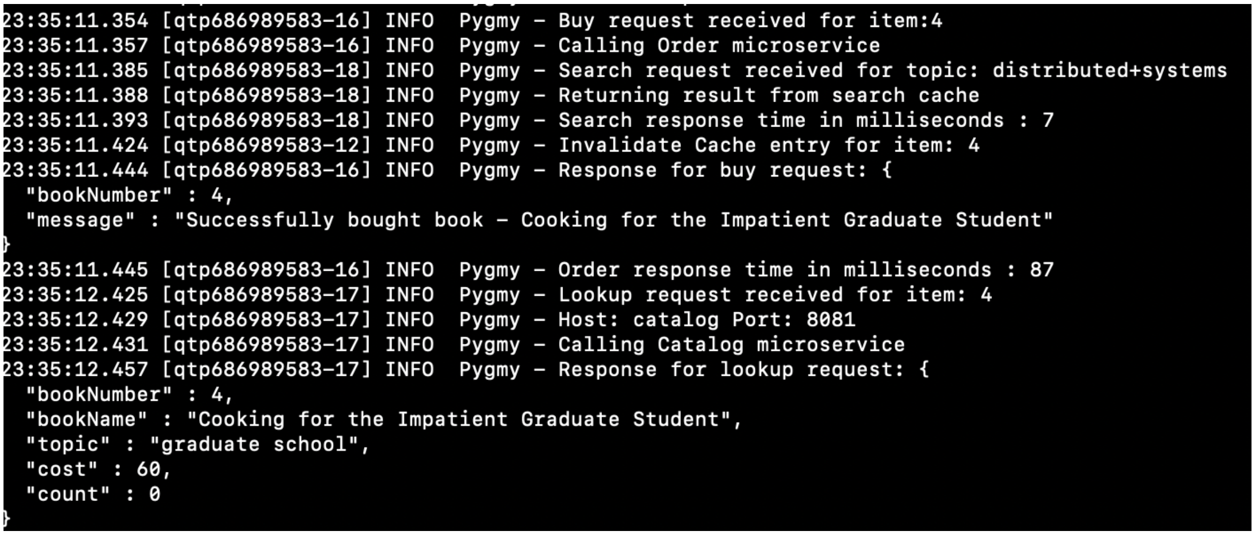
Jagriti Singhal, Bhumika Kalavadia

Date: 30 April, 2021

PART 1: CONSISTENCY AND REPLICATION

CACHING AND CACHE INVALIDATION

Frontend logs



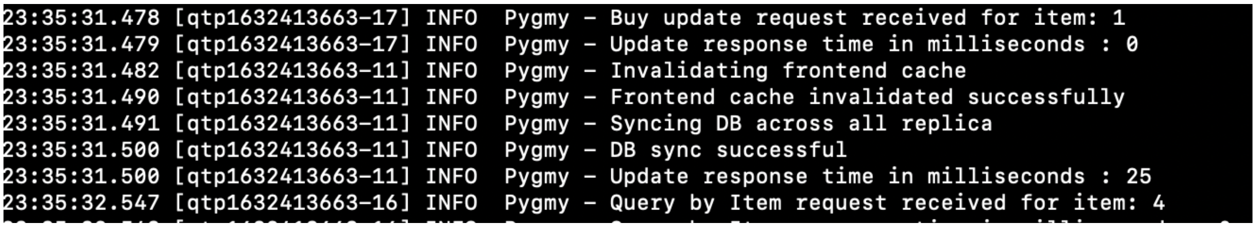
```
23:35:11.354 [qtp686989583-16] INFO  Pygmy - Buy request received for item:4
23:35:11.357 [qtp686989583-16] INFO  Pygmy - Calling Order microservice
23:35:11.385 [qtp686989583-18] INFO  Pygmy - Search request received for topic: distributed+systems
23:35:11.388 [qtp686989583-18] INFO  Pygmy - Returning result from search cache
23:35:11.393 [qtp686989583-18] INFO  Pygmy - Search response time in milliseconds : 7
23:35:11.424 [qtp686989583-12] INFO  Pygmy - Invalidate Cache entry for item: 4
23:35:11.444 [qtp686989583-16] INFO  Pygmy - Response for buy request: {
  "bookNumber" : 4,
  "message" : "Successfully bought book - Cooking for the Impatient Graduate Student"
}
23:35:11.445 [qtp686989583-16] INFO  Pygmy - Order response time in milliseconds : 87
23:35:12.425 [qtp686989583-17] INFO  Pygmy - Lookup request received for item: 4
23:35:12.429 [qtp686989583-17] INFO  Pygmy - Host: catalog Port: 8081
23:35:12.431 [qtp686989583-17] INFO  Pygmy - Calling Catalog microservice
23:35:12.457 [qtp686989583-17] INFO  Pygmy - Response for lookup request: {
  "bookNumber" : 4,
  "bookName" : "Cooking for the Impatient Graduate Student",
  "topic" : "graduate school",
  "cost" : 60,
  "count" : 0
}
```

Here we can see the following -

1. Buy request received for item 4
2. Search request for topic
3. Search request served through **cache**
4. **Invalidate cache** for item 4
5. Lookup request for item 4
6. Lookup request served through catalog server

DB CONSISTENCY

Catalog logs



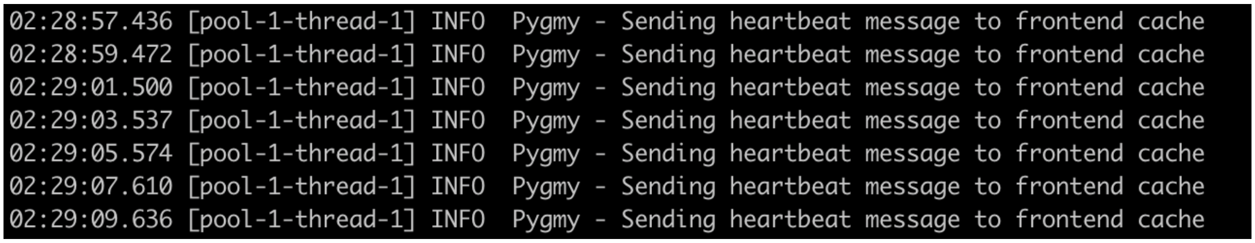
```
23:35:31.478 [qtp1632413663-17] INFO  Pygmy - Buy update request received for item: 1
23:35:31.479 [qtp1632413663-17] INFO  Pygmy - Update response time in milliseconds : 0
23:35:31.482 [qtp1632413663-11] INFO  Pygmy - Invalidating frontend cache
23:35:31.490 [qtp1632413663-11] INFO  Pygmy - Frontend cache invalidated successfully
23:35:31.491 [qtp1632413663-11] INFO  Pygmy - Syncing DB across all replica
23:35:31.500 [qtp1632413663-11] INFO  Pygmy - DB sync successful
23:35:31.500 [qtp1632413663-11] INFO  Pygmy - Update response time in milliseconds : 25
23:35:32.547 [qtp1632413663-16] INFO  Pygmy - Query by Item request received for item: 4
```

On receiving a buy request, the catalog server does the following

1. Invalidates front end cache
2. Sync DB across replica

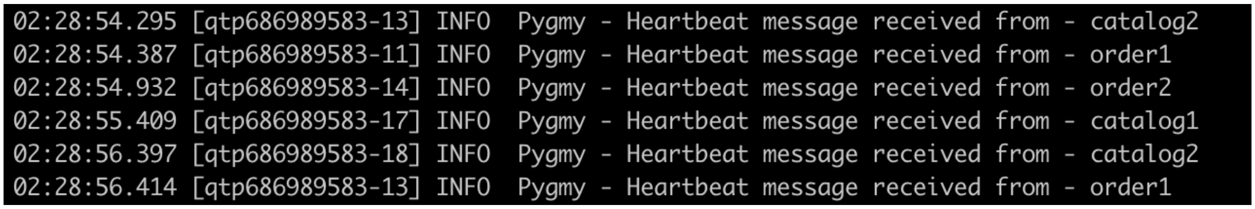
PART 3: Fault Tolerance
Failure Detection: Heartbeat Protocol

Backend servers - catalog1, catalog2, order1 and order2 will be sending heartbeat messages to the frontend server every 2 seconds.

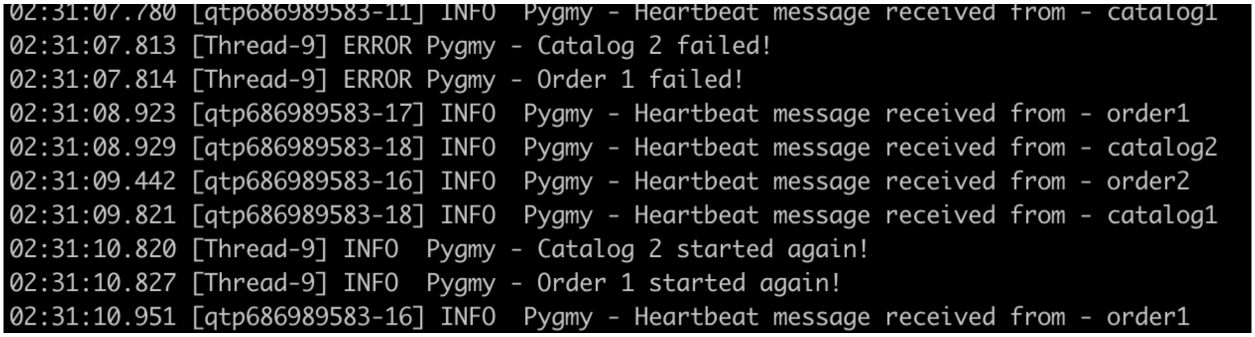


```
02:28:57.436 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:28:59.472 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:29:01.500 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:29:03.537 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:29:05.574 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:29:07.610 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:29:09.636 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
```

Frontend server - checks if heartbeat message is received from backend servers within 3 seconds, else marks server as FAILED. If it gets the response before timeout, it keeps the server in RUNNING state.

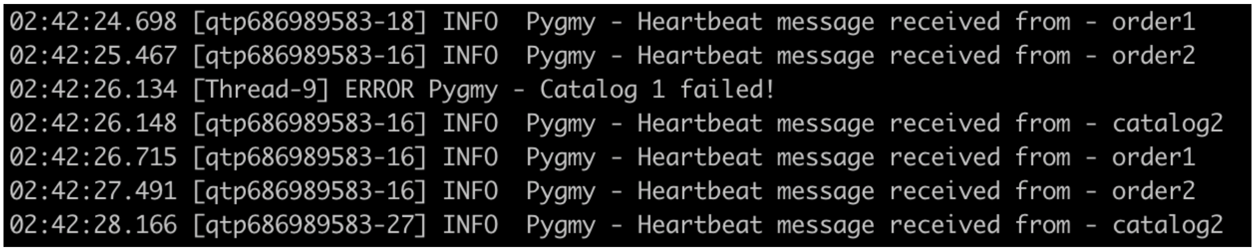


```
02:28:54.295 [qtp686989583-13] INFO  Pygmy - Heartbeat message received from - catalog2
02:28:54.387 [qtp686989583-11] INFO  Pygmy - Heartbeat message received from - order1
02:28:54.932 [qtp686989583-14] INFO  Pygmy - Heartbeat message received from - order2
02:28:55.409 [qtp686989583-17] INFO  Pygmy - Heartbeat message received from - catalog1
02:28:56.397 [qtp686989583-18] INFO  Pygmy - Heartbeat message received from - catalog2
02:28:56.414 [qtp686989583-13] INFO  Pygmy - Heartbeat message received from - order1
```



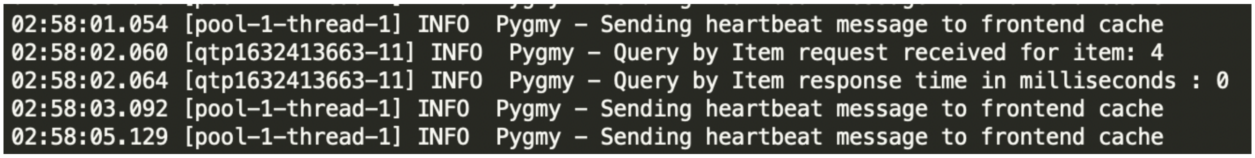
```
02:31:07.780 [qtp686989583-11] INFO  Pygmy - Heartbeat message received from - catalog1
02:31:07.813 [Thread-9] ERROR Pygmy - Catalog 2 failed!
02:31:07.814 [Thread-9] ERROR Pygmy - Order 1 failed!
02:31:08.923 [qtp686989583-17] INFO  Pygmy - Heartbeat message received from - order1
02:31:08.929 [qtp686989583-18] INFO  Pygmy - Heartbeat message received from - catalog2
02:31:09.442 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - order2
02:31:09.821 [qtp686989583-18] INFO  Pygmy - Heartbeat message received from - catalog1
02:31:10.820 [Thread-9] INFO  Pygmy - Catalog 2 started again!
02:31:10.827 [Thread-9] INFO  Pygmy - Order 1 started again!
02:31:10.951 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - order1
```

Fault Tolerance and Recovery
Testcase: Catalog1 server was stopped after some time during execution. Frontend server detected it and marked its status as FAILED.



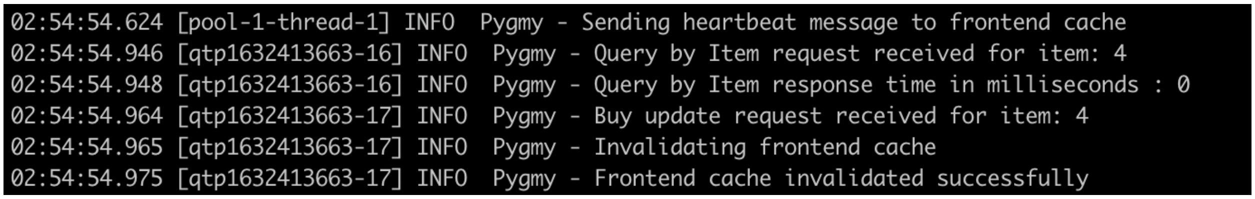
```
02:42:24.698 [qtp686989583-18] INFO  Pygmy - Heartbeat message received from - order1
02:42:25.467 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - order2
02:42:26.134 [Thread-9] ERROR Pygmy - Catalog 1 failed!
02:42:26.148 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - catalog2
02:42:26.715 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - order1
02:42:27.491 [qtp686989583-16] INFO  Pygmy - Heartbeat message received from - order2
02:42:28.166 [qtp686989583-27] INFO  Pygmy - Heartbeat message received from - catalog2
```

All the subsequent requests are sent to catalog server 2. Client sends a buy request to order server 2. Order server checks the availability of the item by calling catalog server 2. Later it sends the update request to catalog server 2 for db updation.
Catalog server 2: Lookup



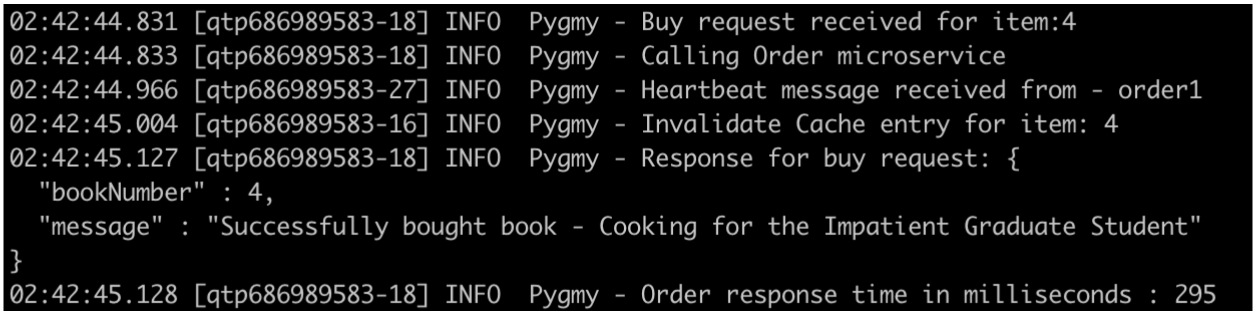
```
02:58:01.054 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:58:02.060 [qtp1632413663-11] INFO  Pygmy - Query by Item request received for item: 4
02:58:02.064 [qtp1632413663-11] INFO  Pygmy - Query by Item response time in milliseconds : 0
02:58:03.092 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:58:05.129 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
```

Catalog server 2: Buy



```
02:54:54.624 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:54:54.946 [qtp1632413663-16] INFO  Pygmy - Query by Item request received for item: 4
02:54:54.948 [qtp1632413663-16] INFO  Pygmy - Query by Item response time in milliseconds : 0
02:54:54.964 [qtp1632413663-17] INFO  Pygmy - Buy update request received for item: 4
02:54:54.965 [qtp1632413663-17] INFO  Pygmy - Invalidating frontend cache
02:54:54.975 [qtp1632413663-17] INFO  Pygmy - Frontend cache invalidated successfully
```

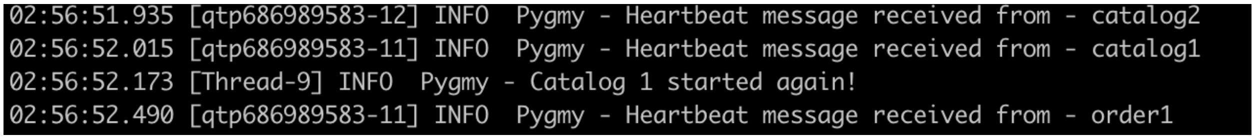
Frontend: Buy



```
02:42:44.831 [qtp686989583-18] INFO  Pygmy - Buy request received for item:4
02:42:44.833 [qtp686989583-18] INFO  Pygmy - Calling Order microservice
02:42:44.966 [qtp686989583-27] INFO  Pygmy - Heartbeat message received from - order1
02:42:45.004 [qtp686989583-16] INFO  Pygmy - Invalidate Cache entry for item: 4
02:42:45.127 [qtp686989583-18] INFO  Pygmy - Response for buy request: {
  "bookNumber" : 4,
  "message" : "Successfully bought book - Cooking for the Impatient Graduate Student"
}
02:42:45.128 [qtp686989583-18] INFO  Pygmy - Order response time in milliseconds : 295
```

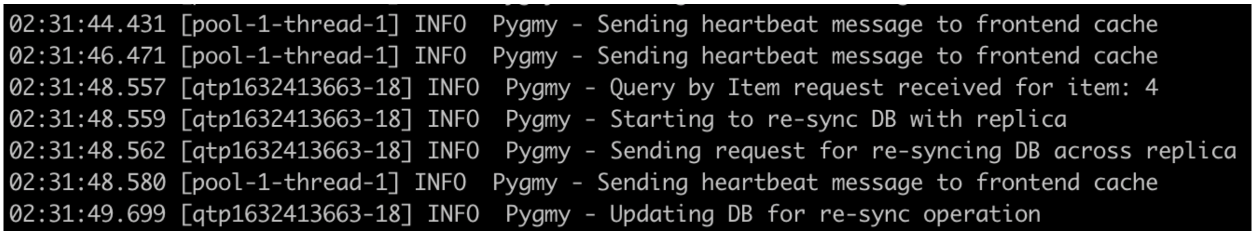
Now, catalog server 1 is started again and we send lookup for item 4 to catalog server 1. As seen below catalog server 1 resync the db with replica before serving the request.

Frontend



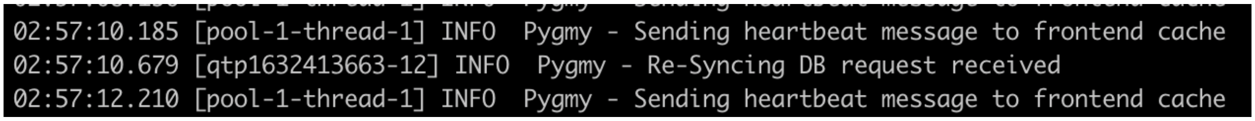
```
02:56:51.935 [qtp686989583-12] INFO  Pygmy - Heartbeat message received from - catalog2
02:56:52.015 [qtp686989583-11] INFO  Pygmy - Heartbeat message received from - catalog1
02:56:52.173 [Thread-9] INFO  Pygmy - Catalog 1 started again!
02:56:52.490 [qtp686989583-11] INFO  Pygmy - Heartbeat message received from - order1
```

Catalog server 1:



```
02:31:44.431 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:31:46.471 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:31:48.557 [qtp1632413663-18] INFO  Pygmy - Query by Item request received for item: 4
02:31:48.559 [qtp1632413663-18] INFO  Pygmy - Starting to re-sync DB with replica
02:31:48.562 [qtp1632413663-18] INFO  Pygmy - Sending request for re-syncing DB across replica
02:31:48.580 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:31:49.699 [qtp1632413663-18] INFO  Pygmy - Updating DB for re-sync operation
```

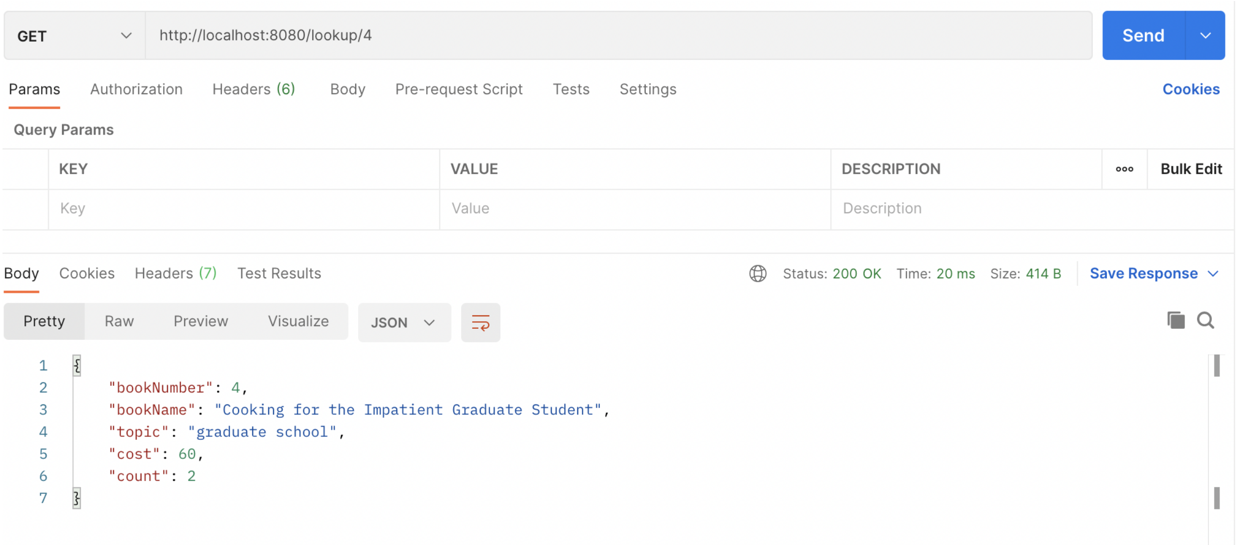
Catalog server 2:



```
02:57:10.185 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
02:57:10.679 [qtp1632413663-12] INFO  Pygmy - Re-Syncing DB request received
02:57:12.210 [pool-1-thread-1] INFO  Pygmy - Sending heartbeat message to frontend cache
```

Resyncing operation:

Catalog Server 1: Number of books of type = 4 is 2.

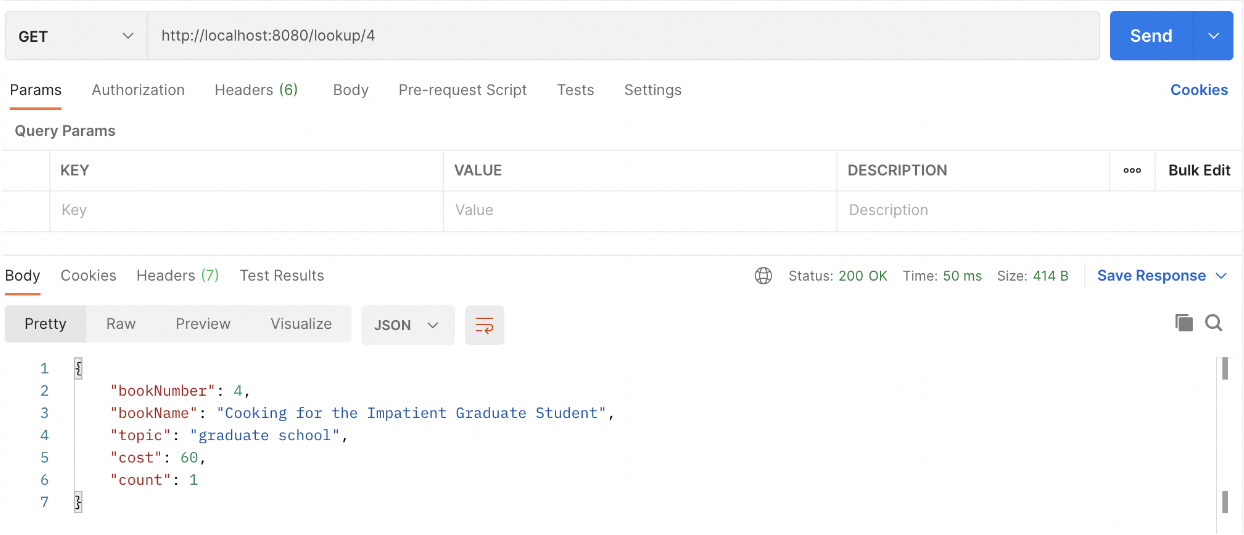


GET http://localhost:8080/lookup/4

Status: 200 OK Time: 20 ms Size: 414 B

```
{
    "bookNumber": 4,
    "bookName": "Cooking for the Impatient Graduate Student",
    "topic": "graduate school",
    "cost": 60,
    "count": 2
}
```

Catalog Server 1 is stopped and then a buy request is made for item 4. This will decrease book count to 1 on catalog server 2 DB. But this change will not be on catalog server 1 as the process has crashed.

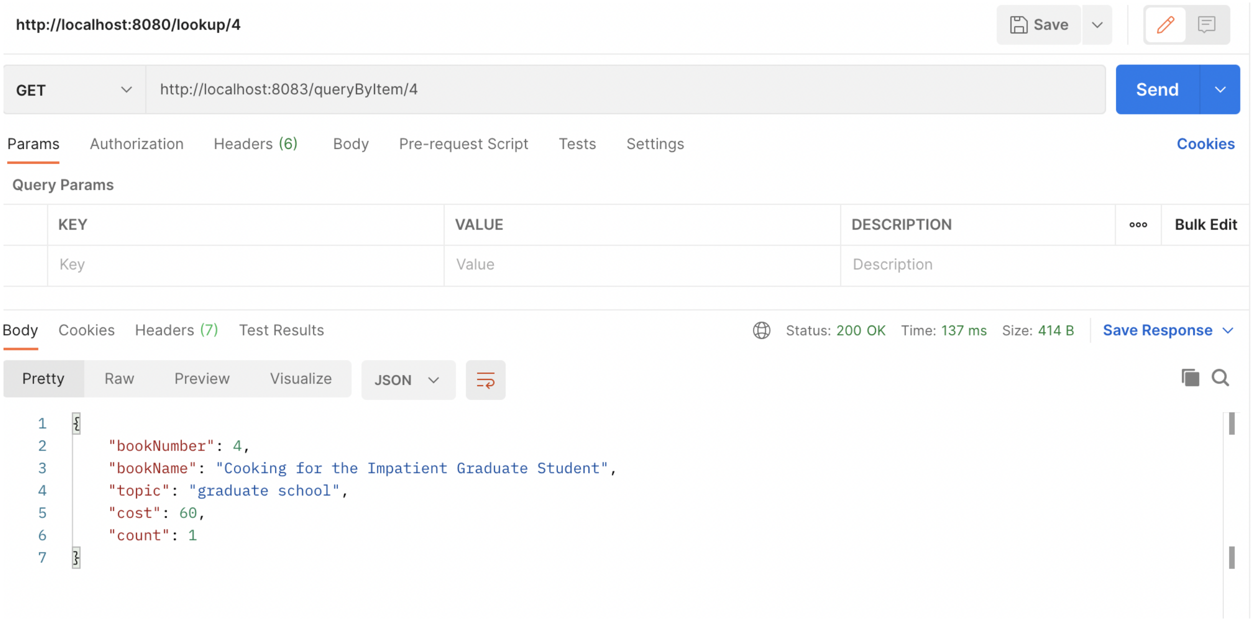


GET http://localhost:8080/lookup/4

Status: 200 OK Time: 50 ms Size: 414 B

```
{
    "bookNumber": 4,
    "bookName": "Cooking for the Impatient Graduate Student",
    "topic": "graduate school",
    "cost": 60,
    "count": 1
}
```

Now starting catalog server 1 again. This will resync the replica db with primary and should reflect book count for type 4 as 1 i.e. same as catalog server 2.



http://localhost:8080/lookup/4

GET http://localhost:8083/queryByItem/4

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 137 ms Size: 414 B

```
{
    "bookNumber": 4,
    "bookName": "Cooking for the Impatient Graduate Student",
    "topic": "graduate school",
    "cost": 60,
    "count": 1
}
```