Distributed Operation Systems

project part-2

Jana Sami Barakeh | 12042410

Lemara Fuad Ali | 120276969

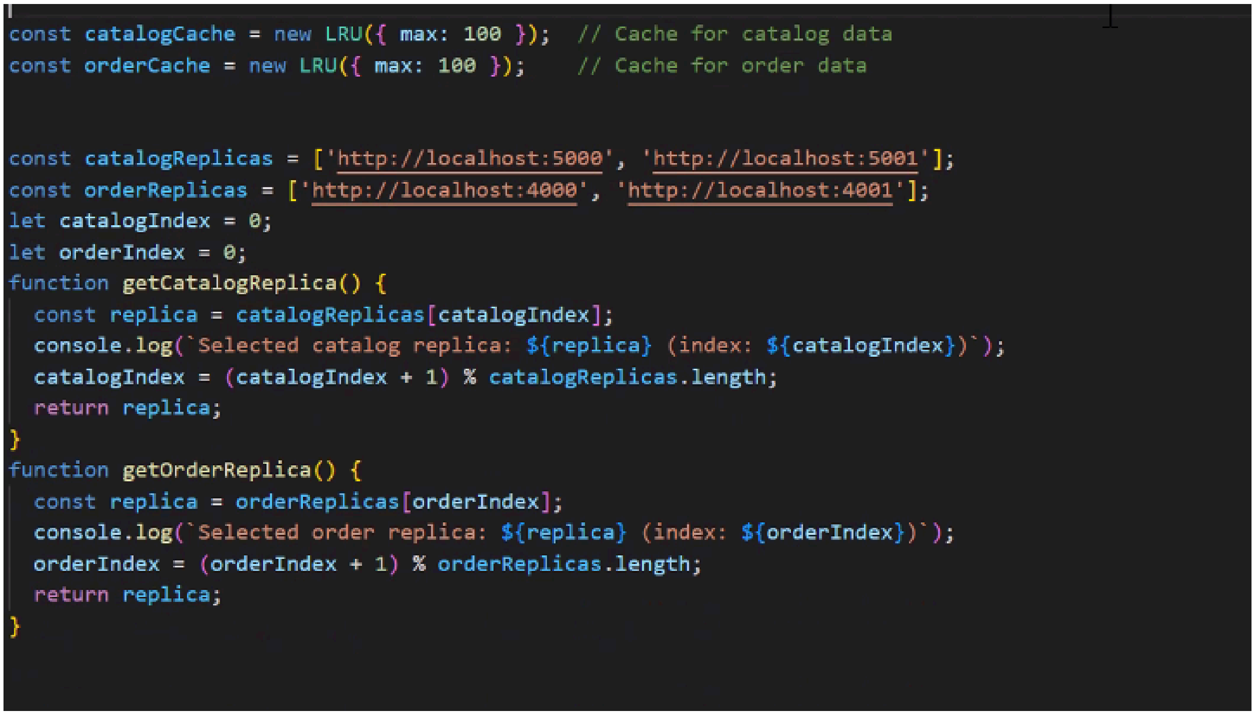
Problem :

We add 3 books in our system and its get big and we have a large number of client so according for this we found that the request from the client take a long time to get the reply so we use two solution :

1) Replication :

We replicated catalog and order server to distribute the client req to replicated server

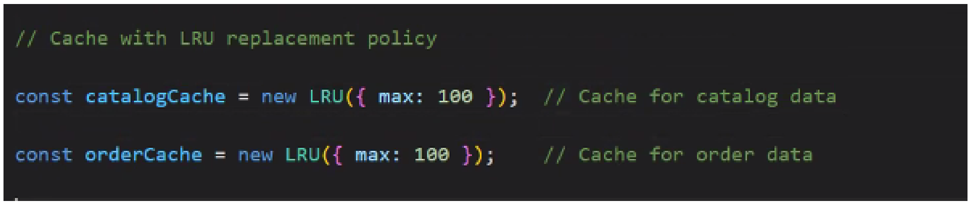
In our system we use 2 replicate for each server and we use round-robin load balancing :



```
const catalogCache = new LRU({ max: 100 });  // Cache for catalog data
const orderCache = new LRU({ max: 100 });    // Cache for order data


const catalogReplicas = ['http://localhost:5000', 'http://localhost:5001'];
const orderReplicas = ['http://localhost:4000', 'http://localhost:4001'];
let catalogIndex = 0;
let orderIndex = 0;
function getCatalogReplica() {
  const replica = catalogReplicas[catalogIndex];
  console.log(`Selected catalog replica: ${replica} (index: ${catalogIndex})`);
  catalogIndex = (catalogIndex + 1) % catalogReplicas.length;
  return replica;
}
function getOrderReplica() {
  const replica = orderReplicas[orderIndex];
  console.log(`Selected order replica: ${replica} (index: ${orderIndex})`);
  orderIndex = (orderIndex + 1) % orderReplicas.length;
  return replica;
}
```

2) Caching :

We make caches to speed up the application process and obtain information.We make a cache for Catalog data and a cache for Order.

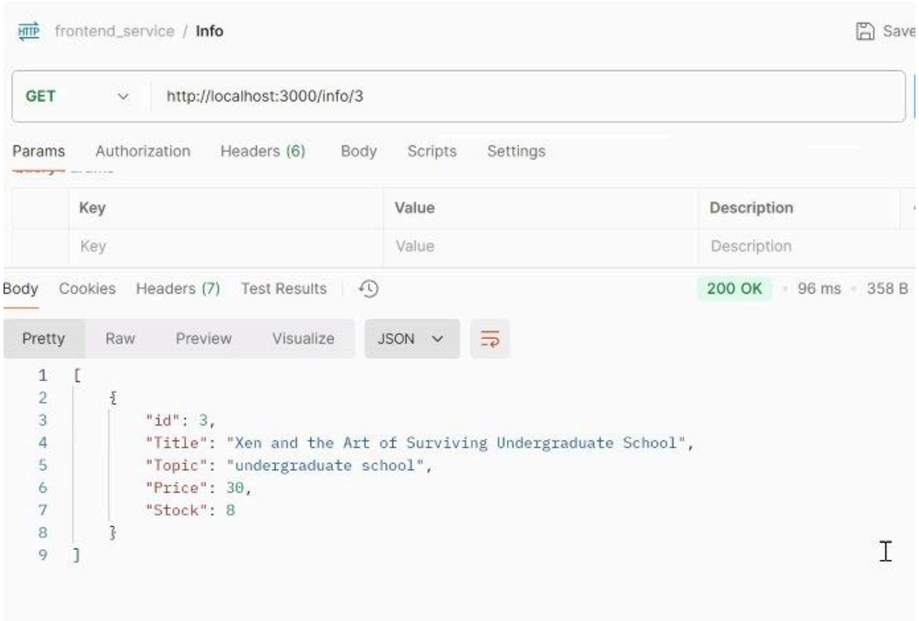


```
// Cache with LRU replacement policy

const catalogCache = new LRU({ max: 100 });  // Cache for catalog data

const orderCache = new LRU({ max: 100 });    // Cache for order data
```

And the results for the cache system is:

Url: http://localhost:3000/info/3.

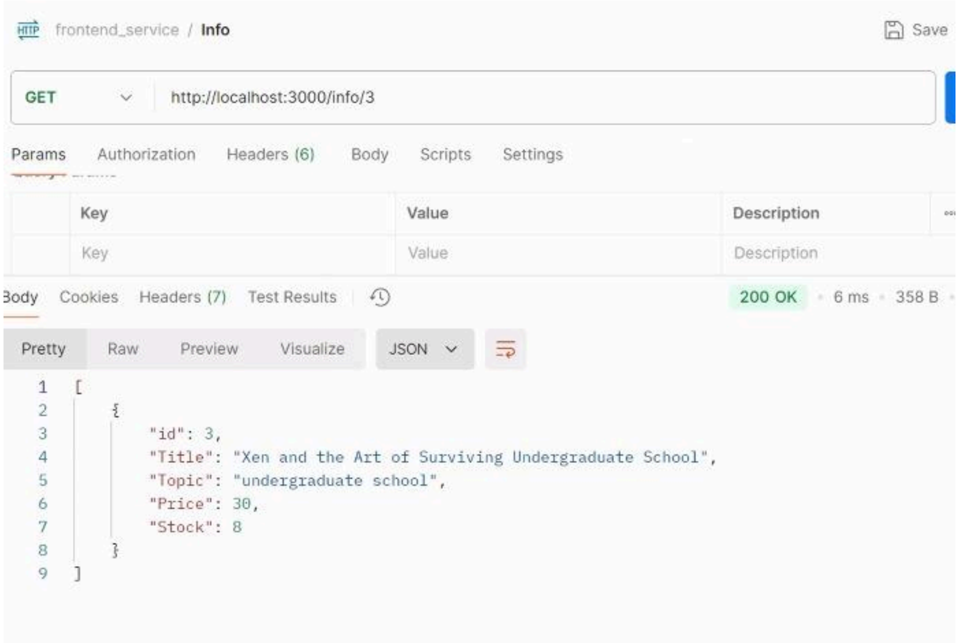
The first request:



http://localhost:3000/info/3

```
[
    {
        "id": 3,
        "Title": "Xen and the Art of Surviving Undergraduate School",
        "Topic": "undergraduate school",
        "Price": 30,
        "Stock": 8
    }
]
```

The second one: → this response came from cache and the time is less than the first call.

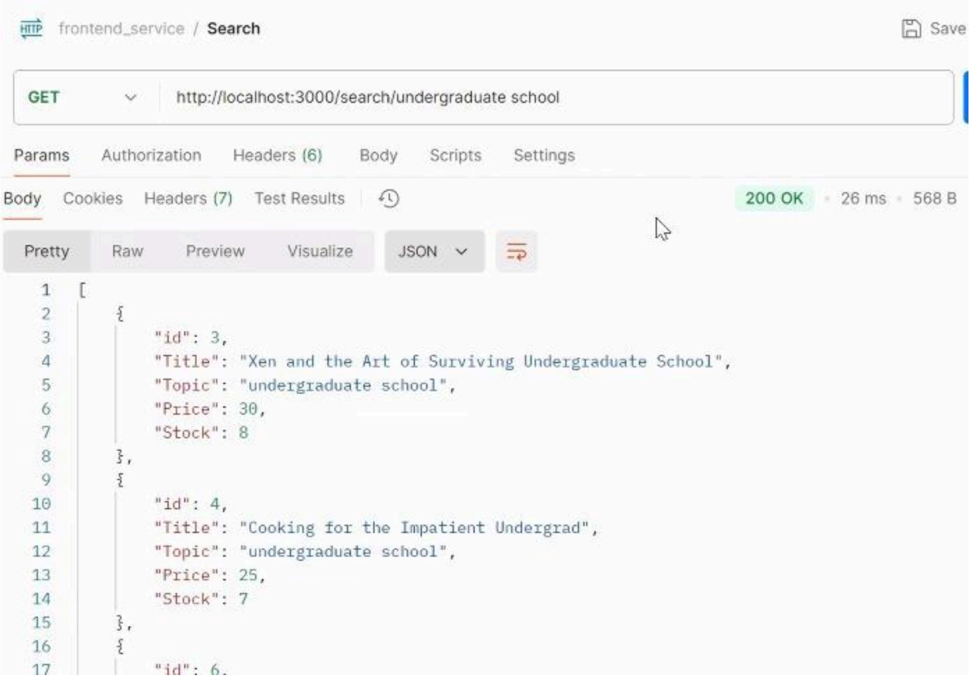


http://localhost:3000/info/3

```
[
    {
        "id": 3,
        "Title": "Xen and the Art of Surviving Undergraduate School",
        "Topic": "undergraduate school",
        "Price": 30,
        "Stock": 8
    }
]
```

Url: http://localhost:3000/search/undergraduate school.

The first request:

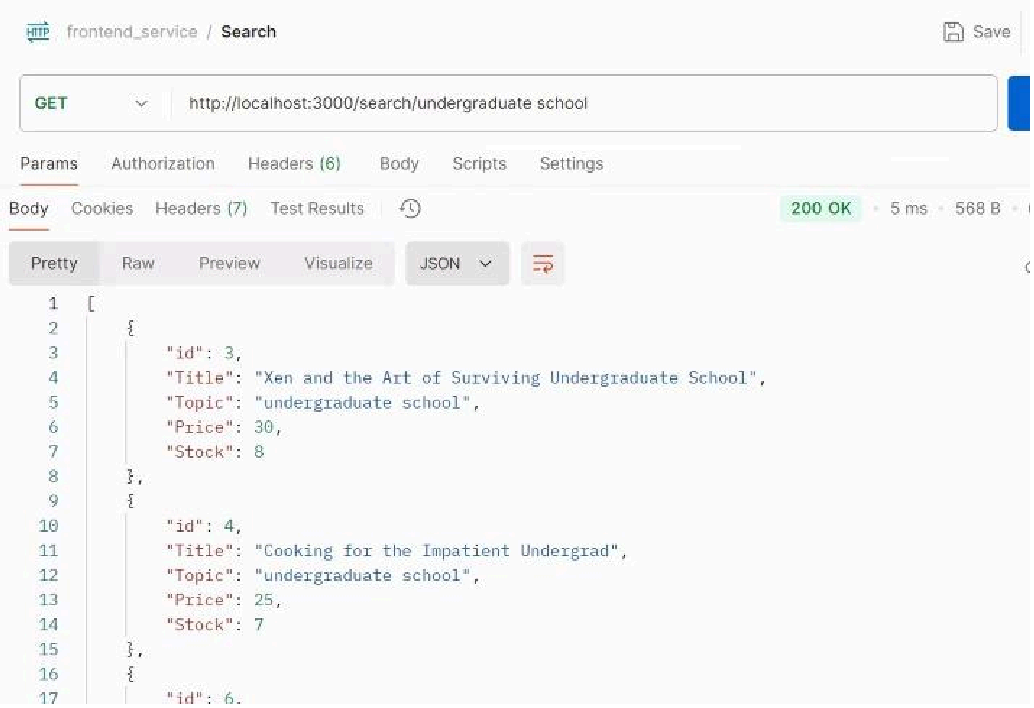


GET http://localhost:3000/search/undergraduate school

200 OK · 26 ms · 568 B

```
[
    {
        "id": 3,
        "Title": "Xen and the Art of Surviving Undergraduate School",
        "Topic": "undergraduate school",
        "Price": 30,
        "Stock": 8
    },
    {
        "id": 4,
        "Title": "Cooking for the Impatient Undergrad",
        "Topic": "undergraduate school",
        "Price": 25,
        "Stock": 7
    },
    {
        "id": 6,
```

The second one: → this response came from cache and the time is less than the first call.



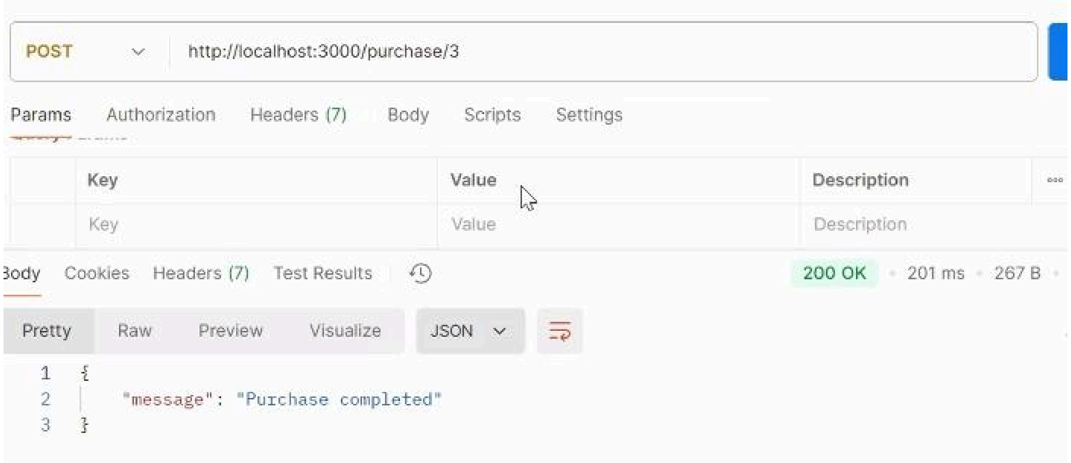
GET http://localhost:3000/search/undergraduate school

200 OK · 5 ms · 568 B

```
[
    {
        "id": 3,
        "Title": "Xen and the Art of Surviving Undergraduate School",
        "Topic": "undergraduate school",
        "Price": 30,
        "Stock": 8
    },
    {
        "id": 4,
        "Title": "Cooking for the Impatient Undergrad",
        "Topic": "undergraduate school",
        "Price": 25,
        "Stock": 7
    },
    {
        "id": 6,
```

Url: http://localhost:3000/purchase/3.

The first request:

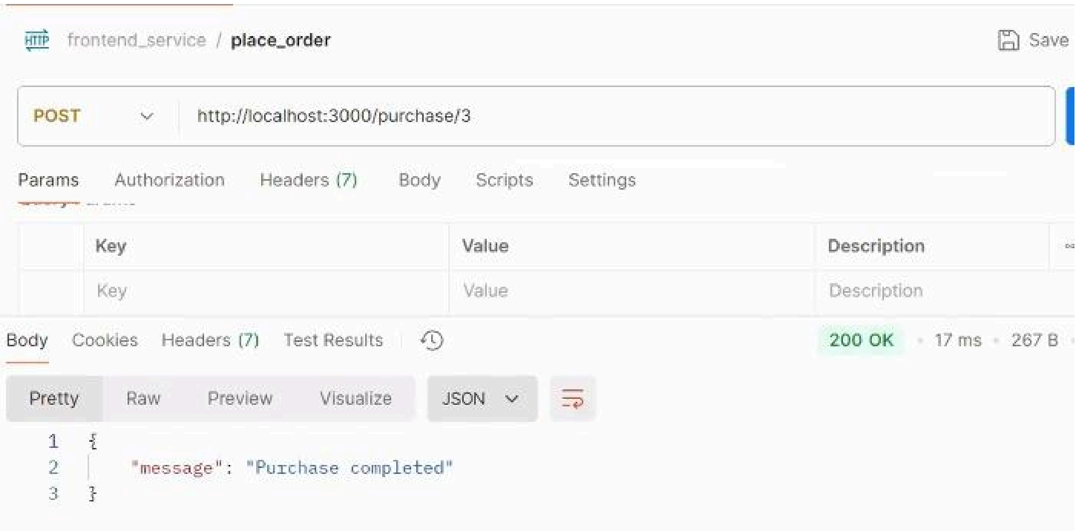


POST http://localhost:3000/purchase/3

200 OK · 201 ms · 267 B

```
{
    "message": "Purchase completed"
}
```

The second one: → this response came from cache and the time is less than the first call.

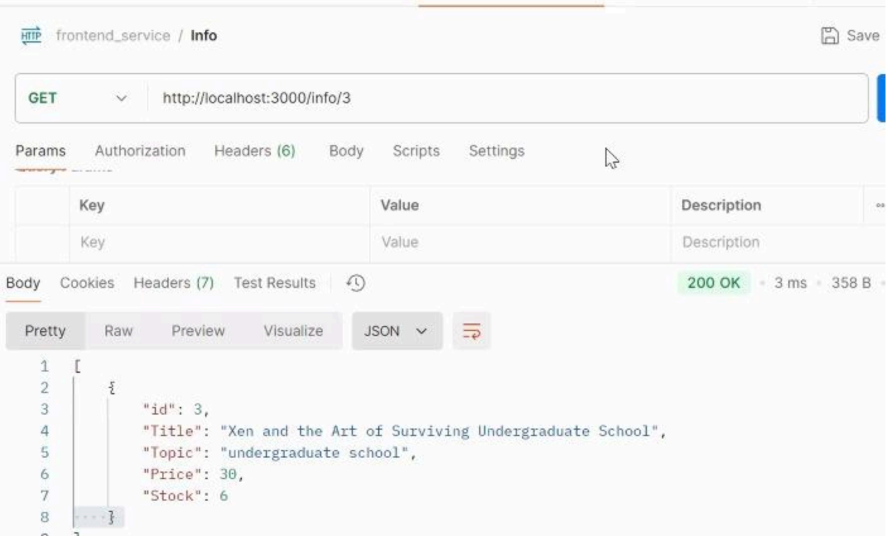


POST http://localhost:3000/purchase/3

200 OK · 17 ms · 267 B

```
{
    "message": "Purchase completed"
}
```

Info from cache after purchase :



GET http://localhost:3000/info/3

```
[
    {
        "id": 3,
        "Title": "Xen and the Art of Surviving Undergraduate School",
        "Topic": "undergraduate school",
        "Price": 30,
        "Stock": 6
    }
```