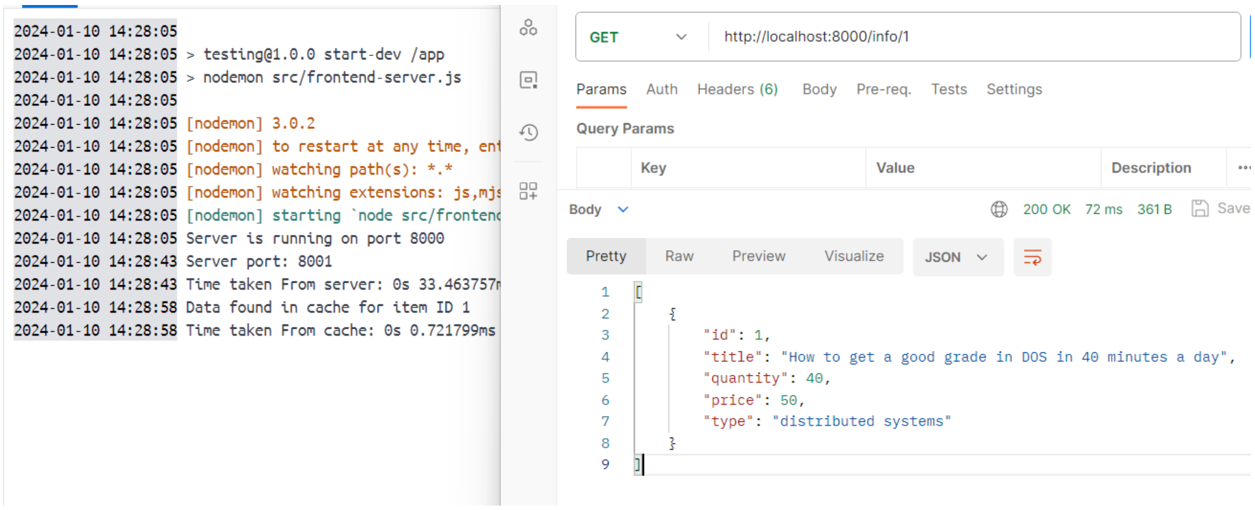
Invalidate the cache experiment

First, we use the info API twice to check if item 1 is saved in the cache.



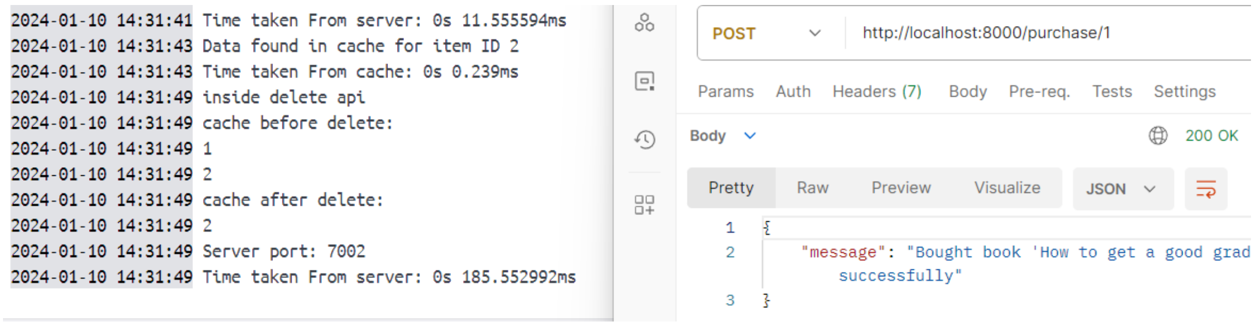
```
2024-01-10 14:28:05
2024-01-10 14:28:05 > testing@1.0.0 start-dev /app
2024-01-10 14:28:05 > nodemon src/frontend-server.js
2024-01-10 14:28:05
2024-01-10 14:28:05 [nodemon] 3.0.2
2024-01-10 14:28:05 [nodemon] to restart at any time, ent
2024-01-10 14:28:05 [nodemon] watching path(s): *.*
2024-01-10 14:28:05 [nodemon] watching extensions: js,mjs
2024-01-10 14:28:05 [nodemon] starting `node src/frontend
2024-01-10 14:28:05 Server is running on port 8000
2024-01-10 14:28:43 Server port: 8001
2024-01-10 14:28:43 Time taken From server: 0s 33.463757r
2024-01-10 14:28:58 Data found in cache for item ID 1
2024-01-10 14:28:58 Time taken From cache: 0s 0.721799ms
```

GET http://localhost:8000/info/1

200 OK 72 ms 361 B

```
[
    {
        "id": 1,
        "title": "How to get a good grade in DOS in 40 minutes a day",
        "quantity": 40,
        "price": 50,
        "type": "distributed systems"
    }
]
```

Then, use the purchase API to buy item 1 and check if the item has been deleted from the cache. In the photo, you can see the cache before deleting item 1 and after it has been deleted:



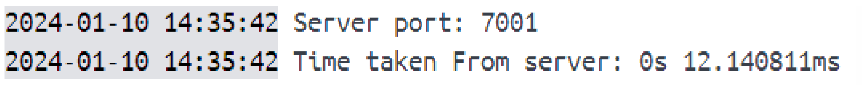
```
2024-01-10 14:31:41 Time taken From server: 0s 11.555594ms
2024-01-10 14:31:43 Data found in cache for item ID 2
2024-01-10 14:31:43 Time taken From cache: 0s 0.239ms
2024-01-10 14:31:49 inside delete api
2024-01-10 14:31:49 cache before delete:
2024-01-10 14:31:49 1
2024-01-10 14:31:49 2
2024-01-10 14:31:49 cache after delete:
2024-01-10 14:31:49 2
2024-01-10 14:31:49 Server port: 7002
2024-01-10 14:31:49 Time taken From server: 0s 185.552992ms
```

POST http://localhost:8000/purchase/1

200 OK

```
{
    "message": "Bought book 'How to get a good grad
        successfully"
}
```

If we use the info API on item 1 again, the response will come from the server instead of the cache:

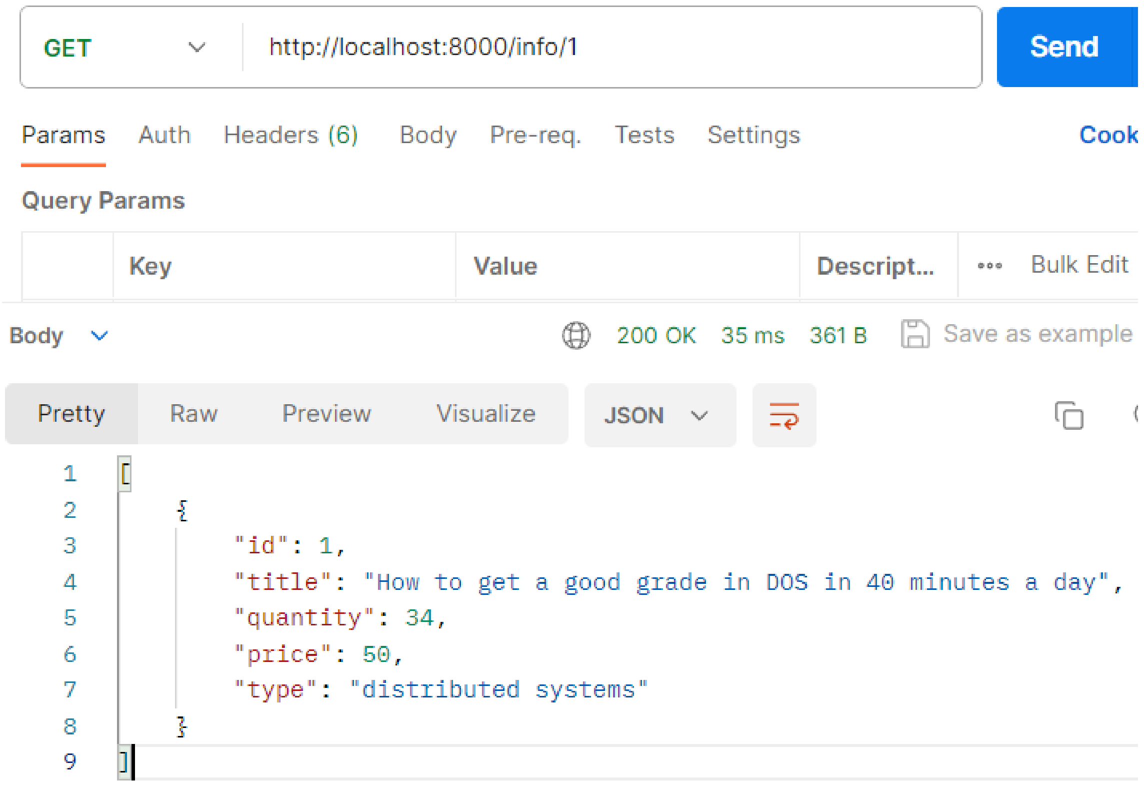


```
2024-01-10 14:35:42 Server port: 7001
2024-01-10 14:35:42 Time taken From server: 0s 12.140811ms
```

Edit Database experiment

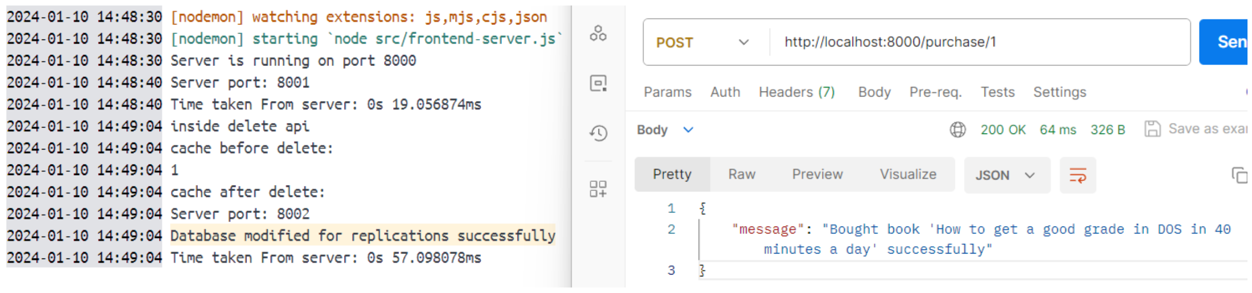
When utilizing the Purchase API and making modifications, such as on the database for replication 1, it is imperative to ensure that the database is promptly updated with the latest data across all replications.

First, use the Info API to check the quantity of the item on Catalog Server 1, which is accessible through port 8001.



```
[
    {
        "id": 1,
        "title": "How to get a good grade in DOS in 40 minutes a day",
        "quantity": 34,
        "price": 50,
        "type": "distributed systems"
    }
]
```

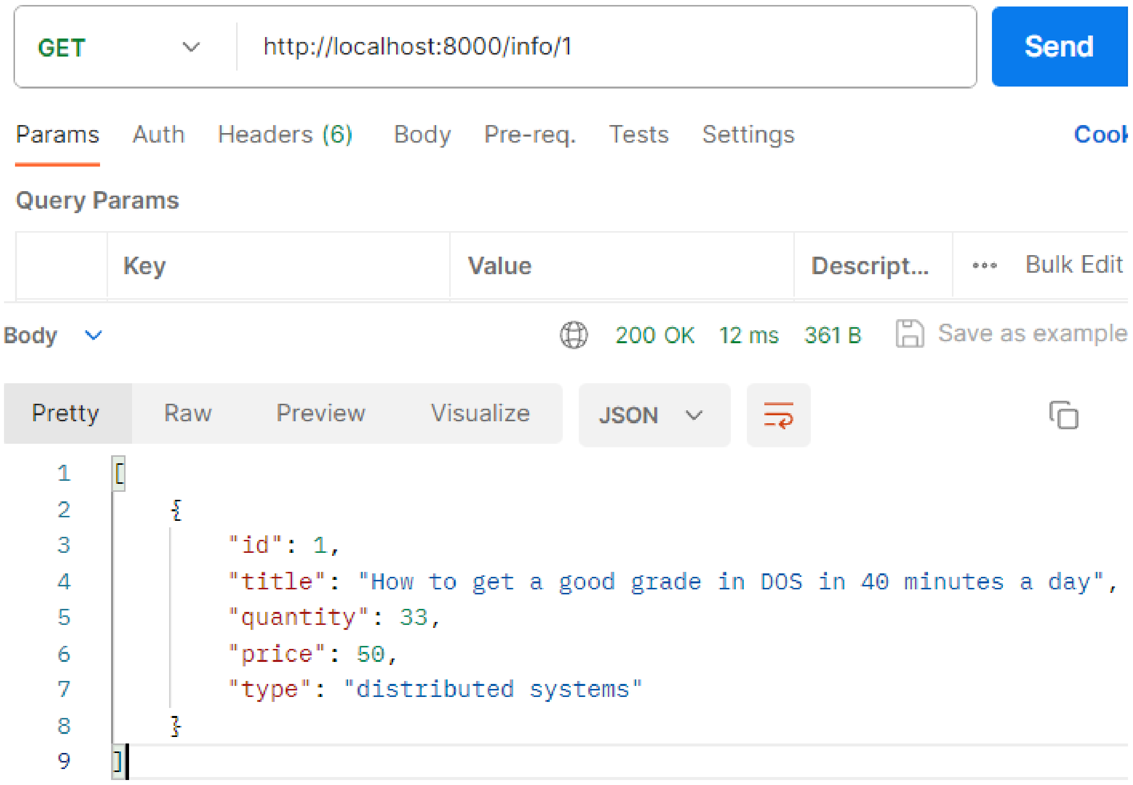
Then, utilize the purchase API to acquire item 1.



```
2024-01-10 14:48:30 [nodemon] watching extensions: js,mjs,cjs,json
2024-01-10 14:48:30 [nodemon] starting `node src/frontend-server.js`
2024-01-10 14:48:30 Server is running on port 8000
2024-01-10 14:48:40 Server port: 8001
2024-01-10 14:48:40 Time taken From server: 0s 19.056874ms
2024-01-10 14:49:04 inside delete api
2024-01-10 14:49:04 cache before delete:
2024-01-10 14:49:04 1
2024-01-10 14:49:04 cache after delete:
2024-01-10 14:49:04 Server port: 8002
2024-01-10 14:49:04 Database modified for replications successfully
2024-01-10 14:49:04 Time taken From server: 0s 57.098078ms
```

```
{
    "message": "Bought book 'How to get a good grade in DOS in 40
        minutes a day' successfully"
}
```

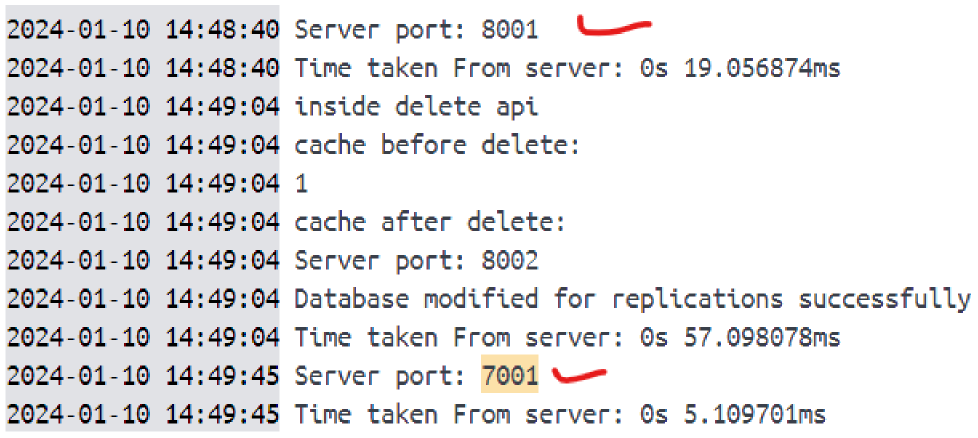
Here we check if the quantity has changed on catalog server 2, port 7001.



```
http://localhost:8000/info/1
```

```
[
    {
        "id": 1,
        "title": "How to get a good grade in DOS in 40 minutes a day",
        "quantity": 33,
        "price": 50,
        "type": "distributed systems"
    }
]
```

Retrieve the first info from catalog server 1 (port 8001) and the second info from catalog server 2 (port 7001).



```
2024-01-10 14:48:40 Server port: 8001
2024-01-10 14:48:40 Time taken From server: 0s 19.056874ms
2024-01-10 14:49:04 inside delete api
2024-01-10 14:49:04 cache before delete:
2024-01-10 14:49:04 1
2024-01-10 14:49:04 cache after delete:
2024-01-10 14:49:04 Server port: 8002
2024-01-10 14:49:04 Database modified for replications successfully
2024-01-10 14:49:04 Time taken From server: 0s 57.098078ms
2024-01-10 14:49:45 Server port: 7001
2024-01-10 14:49:45 Time taken From server: 0s 5.109701ms
```