Submission Information

Option	Description
Name	Lau Jun Hao Benjamin
Matriculation Number	A01840840B
Link to GitHub Repository	(Same as task B) https://github.com/Capeguy/cs3219-otot-b
Instructions	Below
Other Relevant Learnings	null

Task F

Objectives

Successful GET request that retrieves a large amount of data from a local database.

Subsequent successful GET request to the same endpoint that demonstrates substantial performance improvement due to Redis caching

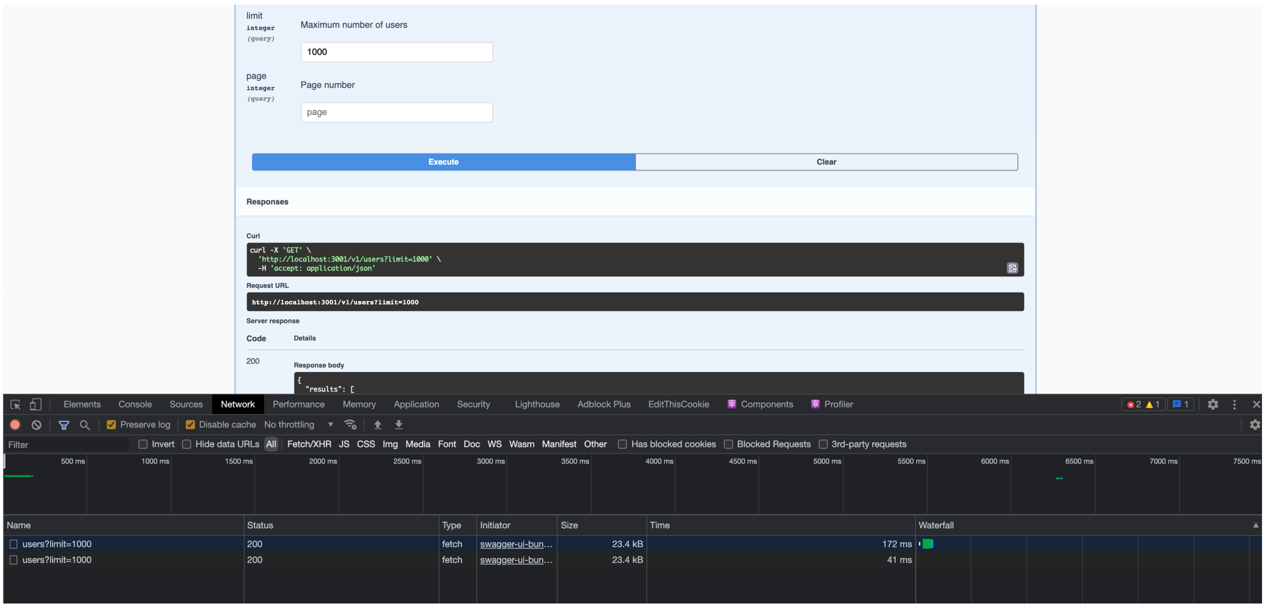
Scenario: Retrieve 1000 users

Excluding Browser to Server latency...

Uncached Response Time: 157.509 ms

Cached response Time: 22.204 ms

Performance Improvement: 85%



limit integer (query) — Maximum number of users: 1000

page integer (query) — Page number: page

```
curl -X 'GET' \
  'http://localhost:3001/v1/users?limit=1000' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3001/v1/users?limit=1000
```

Server response

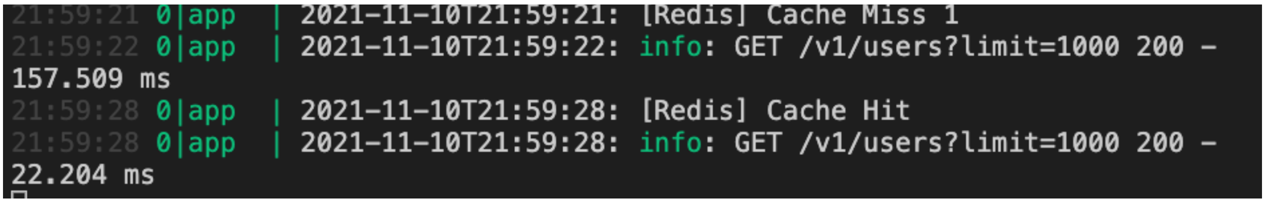
Code	Details
200	Response body

```
{
  "results": [
```

☑ Preserve log ☑ Disable cache No throttling

Filter □ Invert □ Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other □ Has blocked cookies □ Blocked Requests □ 3rd-party requests

Name	Status	Type	Initiator	Size	Time	Waterfall
□ users?limit=1000	200	fetch	swagger-ui-bun…	23.4 kB	172 ms	
□ users?limit=1000	200	fetch	swagger-ui-bun…	23.4 kB	41 ms	



```
21:59:21 0|app  | 2021-11-10T21:59:21: [Redis] Cache Miss 1
21:59:22 0|app  | 2021-11-10T21:59:22: info: GET /v1/users?limit=1000 200 -
157.509 ms
21:59:28 0|app  | 2021-11-10T21:59:28: [Redis] Cache Hit
21:59:28 0|app  | 2021-11-10T21:59:28: info: GET /v1/users?limit=1000 200 -
22.204 ms
```

```
const getUsersCached = catchAsync(async (req, res) => {
  const filter = pick(req.query, ['name', 'role']);
  const options = pick(req.query, ['sortBy', 'limit', 'page']);
  const redisClient = getRedisClient();
  if (redisClient.connected && Object.keys(filter).length === 0) {
    redisClient.get("users", async (err, users)=> {
      if (err) {
        res.json({
            status: "error",
            message: err,
        });
      }
      if (users != null) {
        console.log("[Redis] Cache Hit");
        res.send(JSON.parse(users));
      } else {
        console.log("[Redis] Cache Miss 1");
        const result = await userService.queryUsers(filter, options);
        redisClient.setex("users", 30, JSON.stringify(result));
        res.send(result);
      }
    })
  } else {
    console.log("[Redis] Cache Miss 2");
    const result = await userService.queryUsers(filter, options);
    res.send(result);
  }
});
```