CS6650 Assignments 1
Weinan Shi

1 The URL for git repo

URL: https://github.com/shiwein/CS6650/tree/main/Assignment1

2 Client Design Description

2.1 SkierServlet:

- This is a servlet handling POST requests at the **/skiers/*** endpoint. Its main functionalities include URL validation, JSON request body parsing, and response creation.
- It utilizes the **Gson** library for converting requests and responses to JSON format.
- The **doPost** method validates the request path format, checks if the request body is a valid **LiftRide** object, and returns the appropriate HTTP status code and message based on these validations.
- URL validation is handled by **valPostUrl** and **valSkiersUrl**, ensuring the request path contains the correct resource identifiers.
- The **valLiftRide** method validates the request body to ensure it contains a valid **LiftRide** data structure.

2.2 LiftRide:

- This is a data model class representing a ski lift ride. It has two properties: **time** and **liftID**.
- The class provides getter and setter methods for easy data validation in **SkierServlet**.

2.3 Message:

- This class represents the structure of response messages with a single **message** property, storing feedback for the client.
- The **SkierServlet** uses this class to return the operation status to the client, such as "Data Not Found" or "Lift ride recorded successfully."

2.4 Little's Law and Throughput Prediction

Using Little's Law:

$$L = \lambda \times W \tag{1}$$

where:

- L: average requests in the system (concurrency)
- λ: throughput (requests/second)
- W: average time a request spends

For example, if peak concurrency is 200 and average processing time is 0.1 seconds:

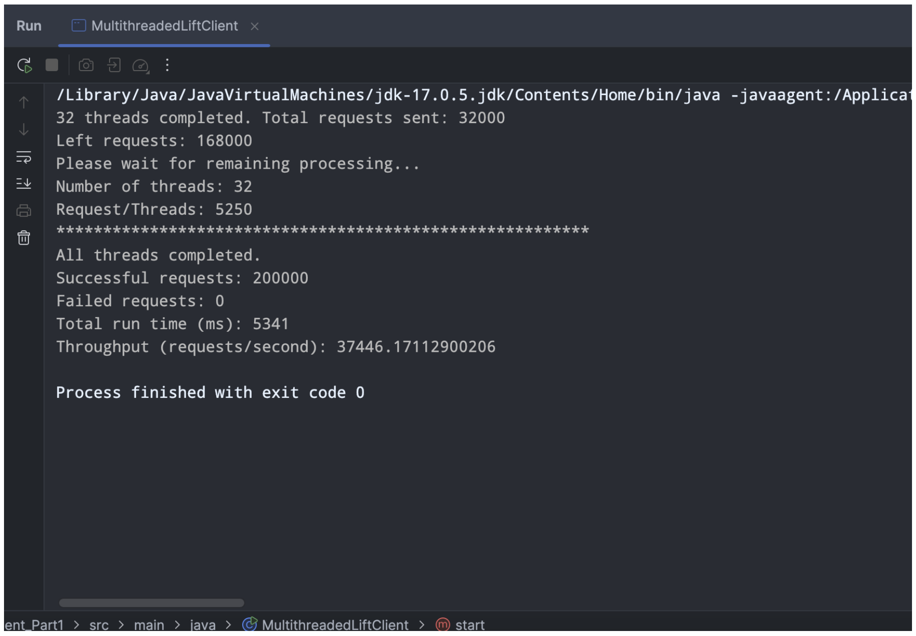
$$\lambda = \frac{200}{0.1} = 2000 \text{ requests/second} \tag{2}$$

This indicates the system can handle around 2000 requests/second at peak, depending on configuration and network conditions.

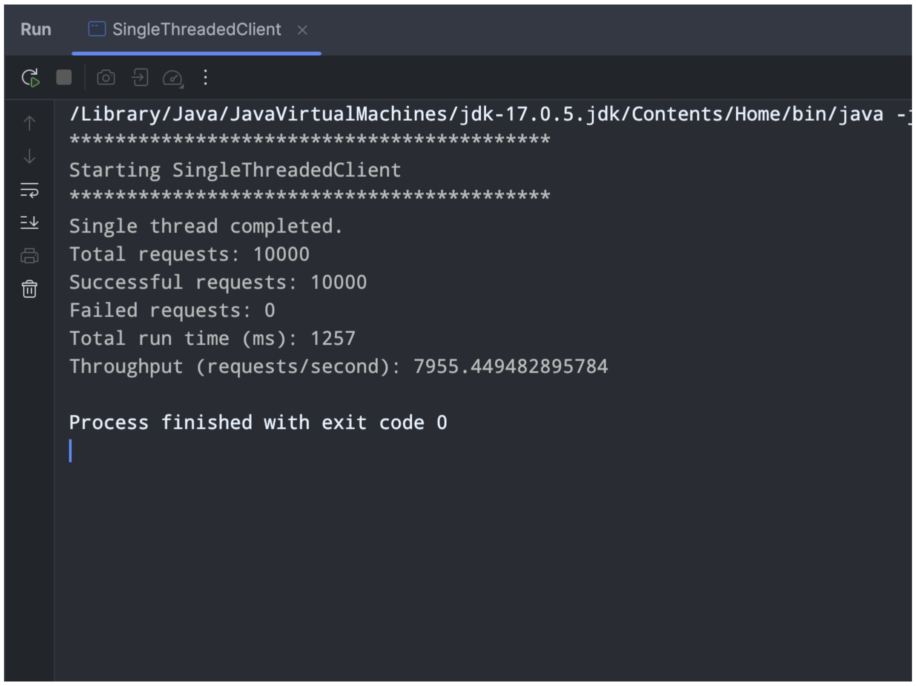
2.5 Summary

My design has a well-organized architecture, with the servlet handling core request logic and data model classes simplifying validation and response creation. Using Little's Law provides a preliminary estimate for throughput, helping in planning system capacity and performance requirements.

3 Client (Part 1)

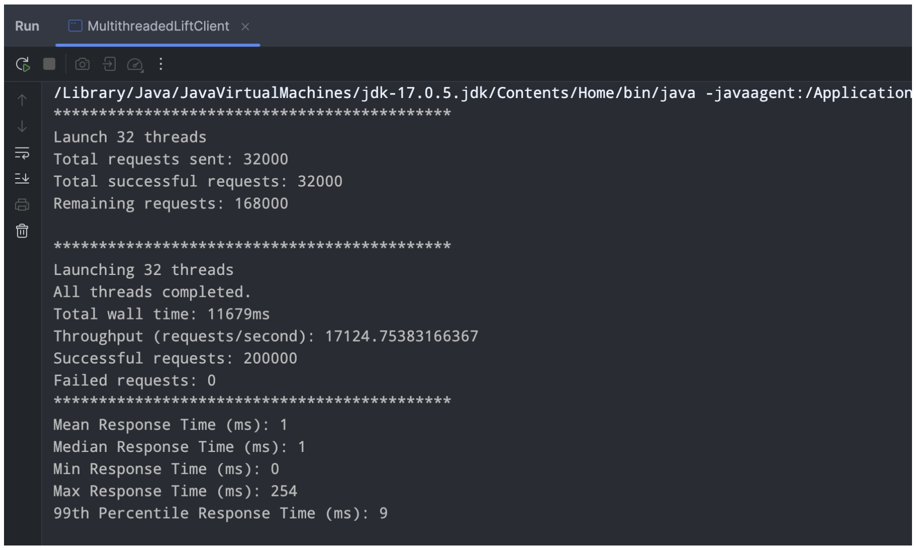

Run
MultithreadedLiftClient
ent_Part1 > src > main > java > MultithreadedLiftClient > start

```
/Library/Java/JavaVirtualMachines/jdk-17.0.5.jdk/Contents/Home/bin/java -javaagent:/Applica
32 threads completed. Total requests sent: 32000
Left requests: 168000
Please wait for remaining processing...
Number of threads: 32
Request/Threads: 5250
********************************************************
All threads completed.
Successful requests: 200000
Failed requests: 0
Total run time (ms): 5341
Throughput (requests/second): 37446.17112900206

Process finished with exit code 0
```

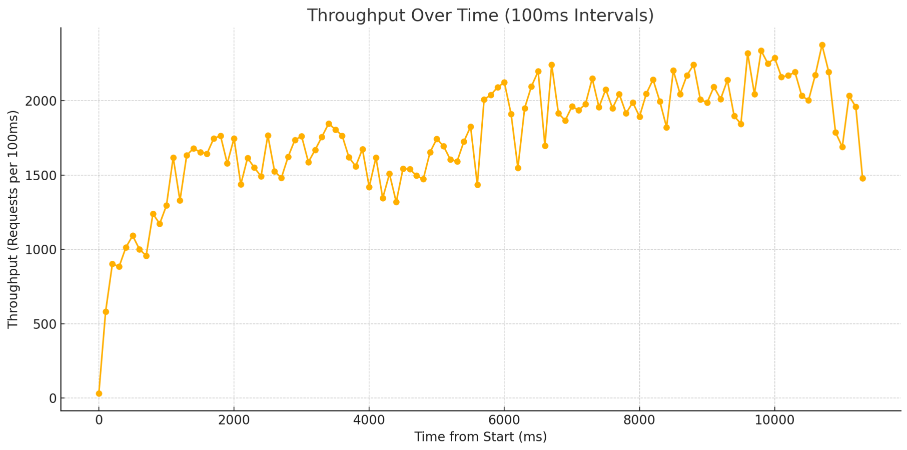


```
/Library/Java/JavaVirtualMachines/jdk-17.0.5.jdk/Contents/Home/bin/java -j
*********************************************
Starting SingleThreadedClient
*********************************************
Single thread completed.
Total requests: 10000
Successful requests: 10000
Failed requests: 0
Total run time (ms): 1257
Throughput (requests/second): 7955.449482895784

Process finished with exit code 0
```

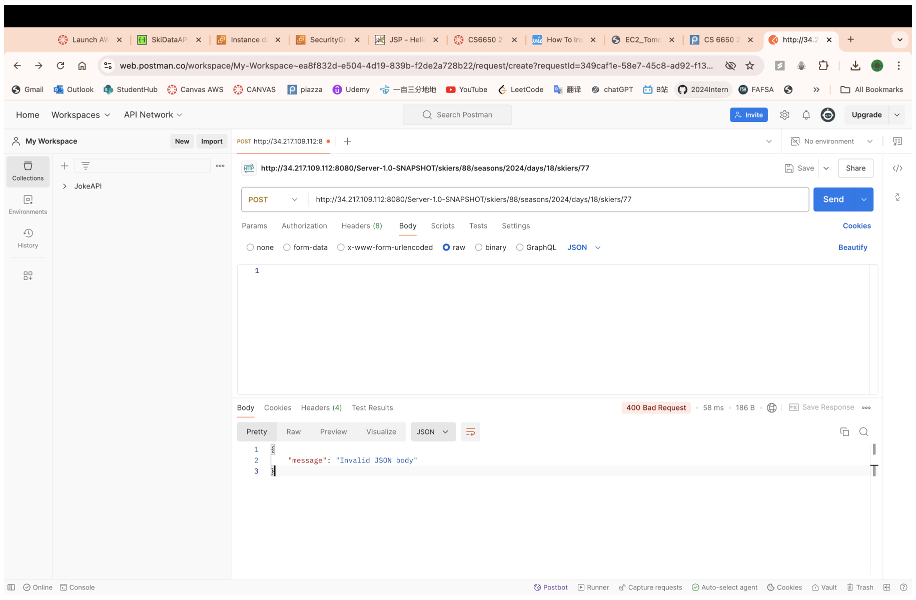
4 Client (Part 2)



```
/Library/Java/JavaVirtualMachines/jdk-17.0.5.jdk/Contents/Home/bin/java -javaagent:/Application
*********************************************
Launch 32 threads
Total requests sent: 32000
Total successful requests: 32000
Remaining requests: 168000

*********************************************
Launching 32 threads
All threads completed.
Total wall time: 11679ms
Throughput (requests/second): 17124.75383166367
Successful requests: 200000
Failed requests: 0
*********************************************
Mean Response Time (ms): 1
Median Response Time (ms): 1
Min Response Time (ms): 0
Max Response Time (ms): 254
99th Percentile Response Time (ms): 9
```



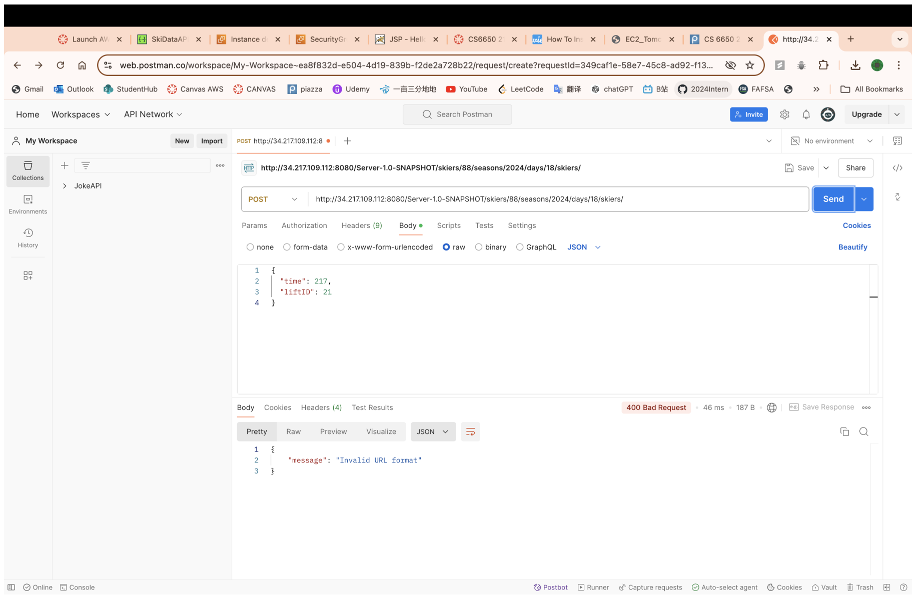
5 Postman

1. 

POST http://34.217.109.112:8080/Server-1.0-SNAPSHOT/skiers/88/seasons/2024/days/18/skiers/77

400 Bad Request · 58 ms · 186 B

```
{
    "message": "Invalid JSON body"
}
```

2. 

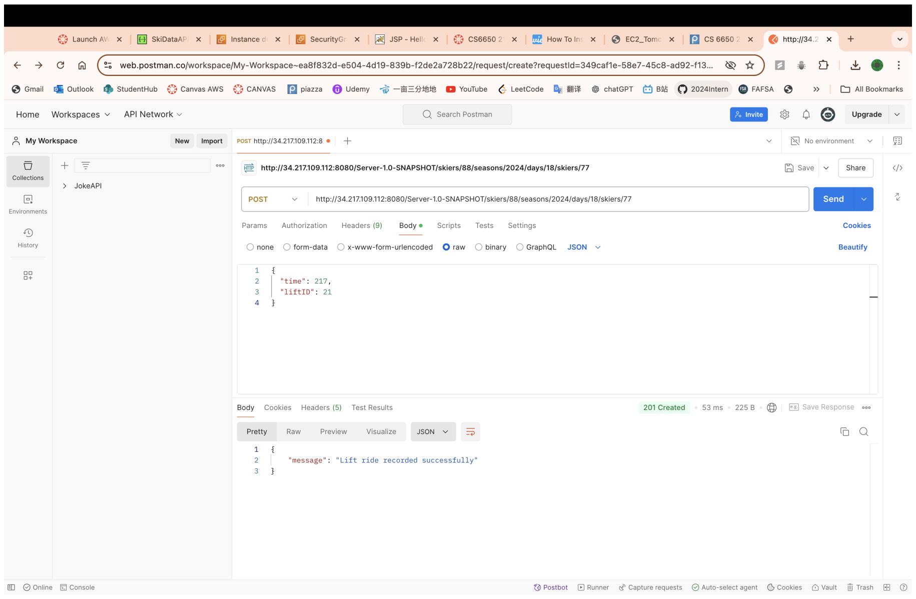
POST http://34.217.109.112:8080/Server-1.0-SNAPSHOT/skiers/88/seasons/2024/days/18/skiers/

raw JSON

```
{
"time": 217,
"liftID": 21
}
```

400 Bad Request · 46 ms · 187 B

```
{
    "message": "Invalid URL format"
}
```

3. 

POST http://34.217.109.112:8080/Server-1.0-SNAPSHOT/skiers/88/seasons/2024/days/18/skiers/77

raw JSON

```
{
"time": 217,
"liftID": 21
}
```

201 Created · 53 ms · 225 B

```
{
    "message": "Lift ride recorded successfully"
}
```