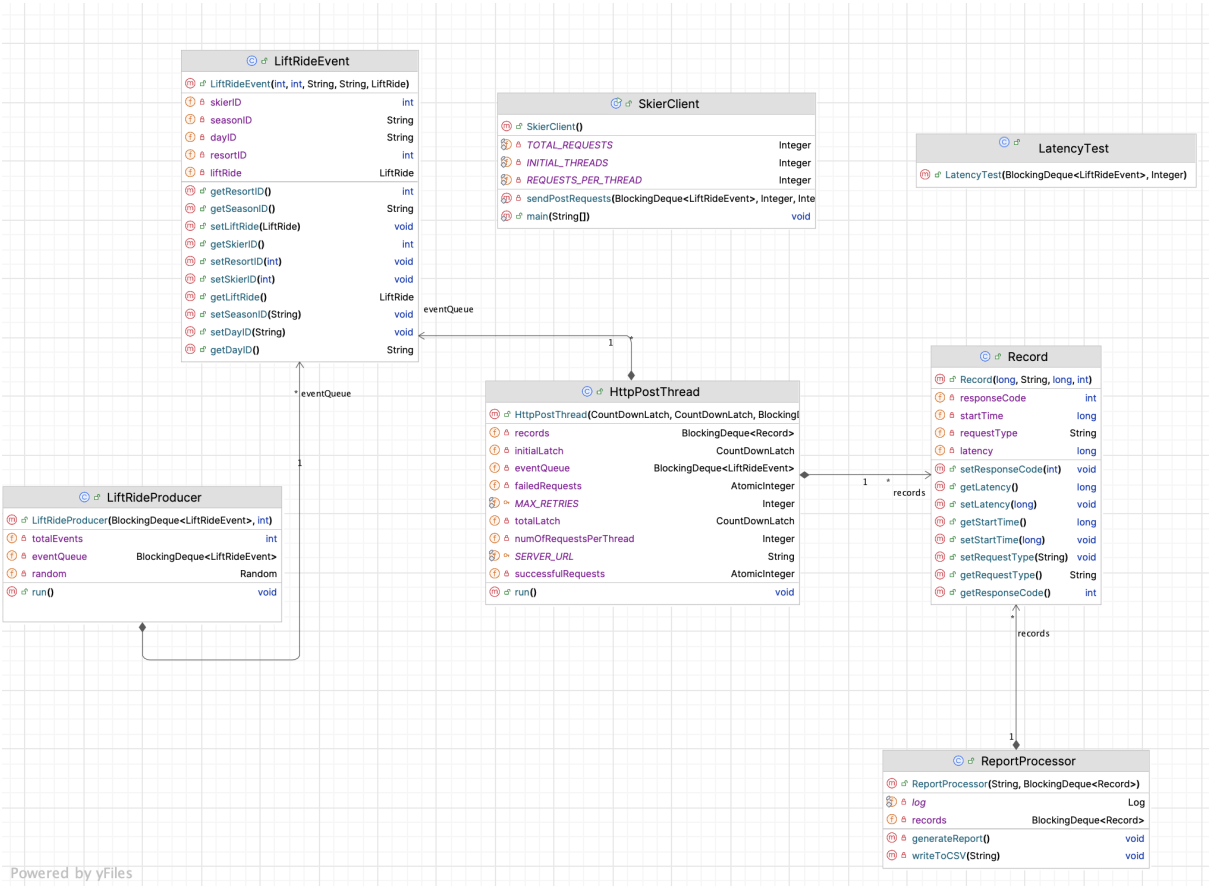
git repo URL

- https://github.com/Syl-Ying/Skier

Client Design



- **A Single Producer**
  - `LiftRideEvent` class: a lift ride event.
  - `LiftRideProducer` class: generate 200k random `LiftRideEvent` instances and store them in a thread-safe queue for the worker threads to consume.

- **Multithreading Consumers**:
  - `HttpPostThread` class: a Runnable class to send given numbers of requests. Each POST request will retry up to 5 times upon receiving 4XX or 5XX errors.
  - `SkierClient` class: the main class
    - First use a single thread to generates 200k events.
    - Then 32 initial threads, each sending 1000 POST requests, will start.
    - Once those threads finish, 168 threads will be created until all 200K requests are sent.
- **Performance Metrics**: Track the number of successful and unsuccessful requests, the total runtime, and calculate throughput (requests per second).
  - `Record` class: a request's status code, starting time…
  - `ReportProcessor` class
    - First calculate related metrics and print to terminal.
    - Then output a CSV file with all the records

Little's Law Throughput Predictions

- Plot of throughput over time (5 points)

  Testing:

```
Testing latency for 1 request and 1 thread:  3 ms
Expected throughput: 333.3333333333333 RequestsPerSecond

Start testing latency.
Start generating lift ride events!

Testing latency for 10000 request and 1 thread:  5 ms
Expected throughput: 2000000.0 RequestsPerSecond
```

  1 request is sent with 1 thread, total time is 3 ms.

  10k requests are sent using 1 thread, total time is 5 ms.

  W - Average response time / Latency: 3ms

  L - Number of requests sent: 10,000

λ = L / W = 10,000 / (3 / 1000) = 3,333,000 rps. so expected throughput is 3,333,000.

Actual:

200k requests are sent using 200 threads.

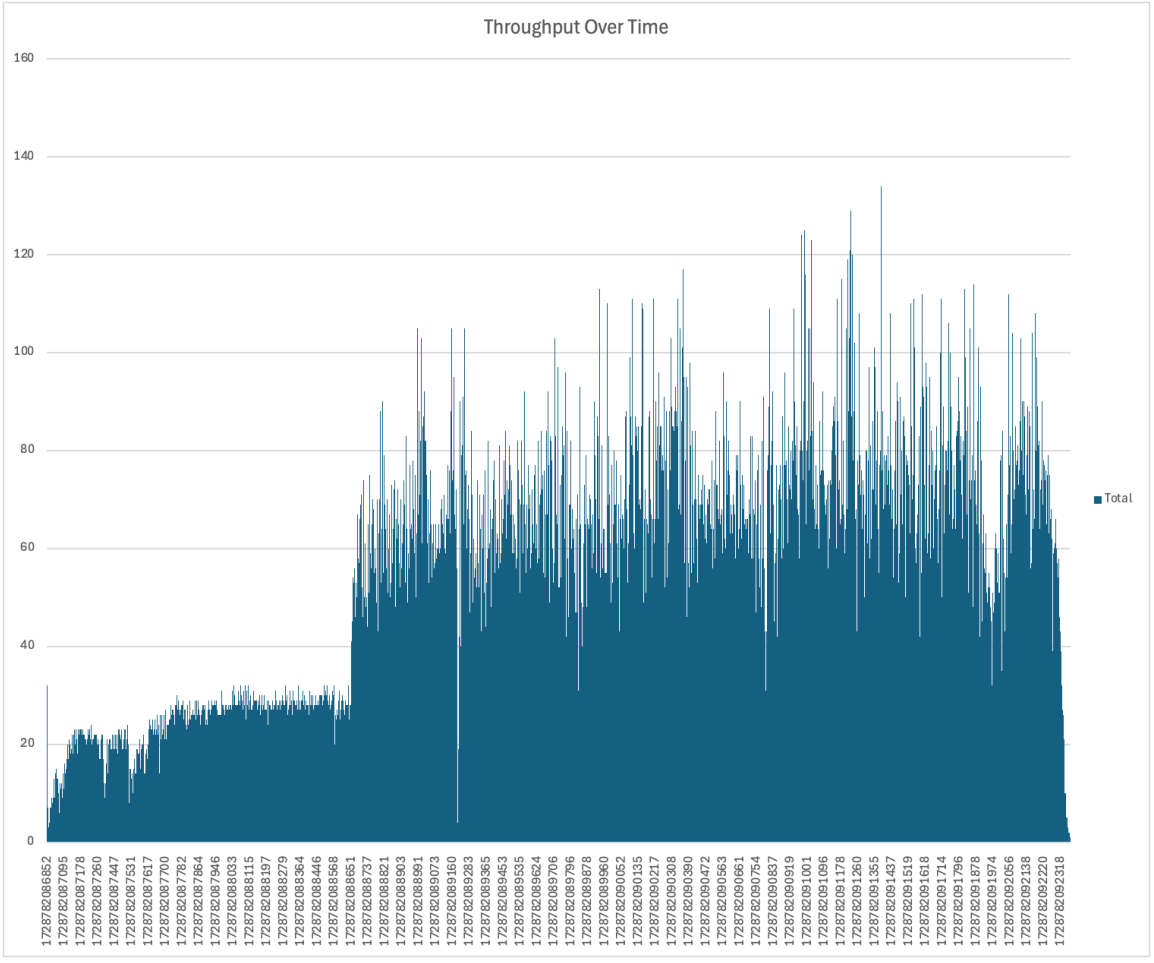
W: Mean response time 5 ms.

N: I have 200 client threads and 200 default Tomcat threads, and hence the maximum is 200 concurrent requests.

Prediction λ = 200 / 0.005 = 40,000 rps

Actual λ = Actual Requests / Actual Time = 200k / 8.515 = 23,488 rps

The difference is caused by waiting for one of the 32 threads to finish before starting the remaining 168 threads in actual settings. The prediction assumes 200 threads are started simultaneously.



Y axis: requests numbers at the specific time. X axis: time

Client (Part 1)

a screenshot of output window with your `wall time` and `throughput`. include the client configuration in terms of the `number of threads used` (print it out in your output window).

```
---------------------- Report Client1 ----------------------
Number of successful requests: 200,000
Number of failed requests: 0
Wall time for all 200k requests using 200 threads: 8,515 ms
Actual throughput: 23,487.96 RequestsPerSecond
----------------------  Report End  ------------------------
```

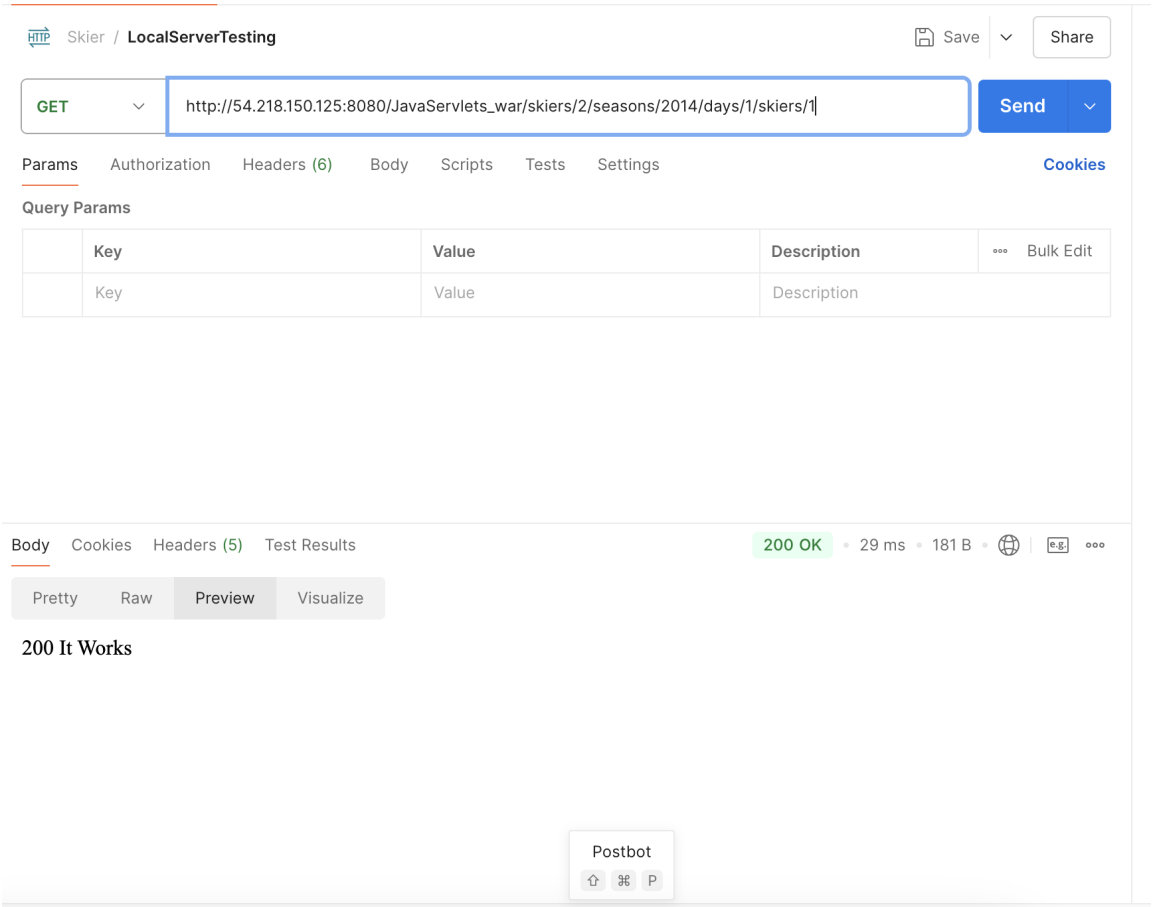
Client (Part 2)

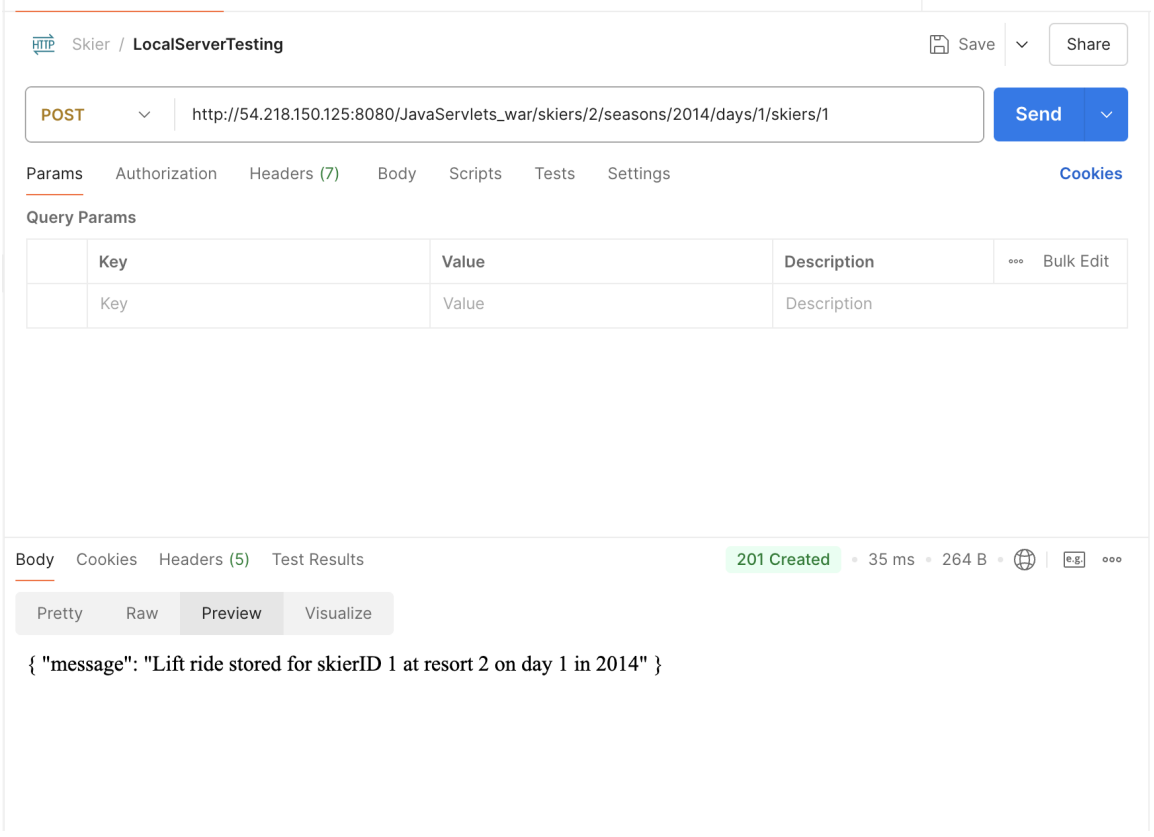
- run the client as per Part 1, showing the output window for each run with the specified `performance` statistics listed at the end.

```
---------------------- Report Client2 ----------------------
Mean response time (millisecs): 5
Median response time (millisecs): 5
p99 (99th percentile) response time: 6
Min response time (millisecs): 0
Max response time (millisecs): 191
Throughput (RPS): 23,755.79
---------------------- Report End --------------------------
```

EC2

- screenshots to show you actually send your requests to the server on EC2 instance: Postman testing page showing the Url, and/or the EC2 instance page.


Skier / LocalServerTesting
Save
Share
GET
http://54.218.150.125:8080/JavaServlets_war/skiers/2/seasons/2014/days/1/skiers/1
Send
Params
Authorization
Headers (6)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
200 OK
29 ms
181 B
Pretty
Raw
Preview
Visualize
200 It Works
Postbot


Skier / LocalServerTesting
Save
Share
POST
http://54.218.150.125:8080/JavaServlets_war/skiers/2/seasons/2014/days/1/skiers/1
Send
Params
Authorization
Headers (7)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
201 Created
35 ms
264 B
Pretty
Raw
Preview
Visualize
{ "message": "Lift ride stored for skierID 1 at resort 2 on day 1 in 2014" }