Testing:

While building the backend, I tested directly from the container to confirm that /postmessage would function

npx loadtest -c 10 -n 100 -m POST -T "application/json" --data ‘{"topic”:”test”,”data”:”something"}’ <http://localhost:8080/postmessage>

Target URL: http://localhost:8080/postmessage

Max requests: 1000

Concurrent clients: 60

Running on cores: 6

Agent: none

Completed requests: 1000

Total errors: 0

Total time: 0.425 s

Mean latency: 23.1 ms

Effective rps: 2353

Percentage of requests served within a certain time

50% 23 ms

90% 26 ms

95% 30 ms

99% 33 ms

100% 39 ms (longest request)

Direct access to the server for loadtest showed a mean latency of 23.1ms, with 39ms being the longest request, with 60 effective concurrent clients 1000 requests were completed within 425ms. Testing with 180 concurrent clients and 1000 requests (-c 15, --cores 12) ran for 547ms, but had higher latency 71ms mean and 119ms max. Testing with 1 concurrent client and 1000 request ran for 767ms with a much lower latency 0.1ms mean and 11ms max.

Running multiple tests of the same request and concurrency, -c 20 –cores 6 had varying results with Effective rps between 2105 and 1733.

From the first loadtest I checked the posting page to view the entries and loaded the 1000 posts I would expect to see. I confirmed with later with

curl -H ‘Content-Type: application/json’ -X GET <http://localhost:8080/messages>

which confirmed the number of posts, checking the posting.html page after building also showed the expected number of posts from the load test.

While building posting.html I would build and test for posting and retrieval, once the posts were working I added the response/reply piece and tested that I could post and add a reply, then add multiple posts and have replies correctly associated with the post they were entered from.

After posts, replies and retrieval were working I started trying to add jqueryui elements, I started with an accordion to allow collapsing the posts. Posting continued to work but replies would send null for the postId.

After attempting to solve this issue I pivoted and added a dialog to add a new post, after which testing showed both /postmessage and /postresponse were receiving the expected information.

CSS elements were added after all required elements were confirmed working.

A screenshot of a computer

AI-generated content may be incorrect.