

Nano Twitter 0.4

What we have done.

- 1) Implemented the test interfaces.
- 2) Follow, search (username and tweet content), profile display, home time line display, home page.
- 3) Avoid page redirection (to avoid query time line too often) and make UI more fluent by Backbone.js and Bootstrap.css
- 4) Deployed to Heroku. Check at <https://nano-twitter-xx.herokuapp.com/>
- 5) Done loader.io test. Interesting!

In the Index page, which displays the recent 50 tweets.

If I get 40 clients per second and the test for 30 seconds, the average response time is 1678ms, and when I get 100 clients per seconds, the average response time goes to over 4.5s.

- 6) Changed our Database schemas

When we actually implemented the functionalities, we find a lot of inconvenience and inefficiency in the old database schemas, and we also find that when implementing test interfaces, it is necessary to delete one specific user. And thus when we have to delete all the associations when a user is deleted, so we add more restriction to the models.

What are the problems and what are the future work

- 1) There is an obvious non-linear time cost rising when I get more and more

clients. The next step is mainly focused on how to optimize the performance.

Maybe it is the time for Redis.

- 2) Other functionalities haven't finished: like hash tag search, reply, and likes.