

note: routes starting with * are of low priority
reference: nano Twitter Functionality

1 Basic path

HTTP GET

- / (the home page for login; if logged in, redirect to user's home timeline)
- /logout (page after the user logs out)
- /register (page for creating a new account)
- /home (show a user's home timeline; if not logged in, redirect to the home page)
- /browse (see the last n tweets by any user)
- /user/<user_id> (see the last n tweet by this user)
- /friends (show whom the user is following)
- /followers (show who are following this user)
- */remove_friend (unfollow a user)

HTTP POST

- /user/<user_id>/tweet (for sending tweets)
- /user/<user_id>/follow (for following other users)
- /user/<user_id>/unfollow (for unfollowing other users)

2 API

HTTP GET

- /api/v1/tweets/<tweet_id> (return tweet with id)
- /api/v1/users/<user_id> (return user information)
- /api/v1/tweets/recent (return the recent n tweets)
- /api/v1/users/<user_id>/tweets (return the user's recent tweets)
- /api/v1/users/<user_id>/friends (return all users ids that the user is following)
- /api/v1/users/<user_id>/followers (return all followers ids of the user)

HTTP POST

- /api/v1/tweets/update (send a tweet)
- /api/v1/users/follow (follow a user)
- */api/v1/users/unfollow (unfollow a user)

3 Test interface

In general, `<user_id>` = “testuser”

- `/test/reset/all` (delete all users, tweets, follows)
- `/test/reset/<user_id>` (delete and recreate this user)
- `/test/status` (report how many users, follows, and tweets are there, etc.)
- `/test/users/create?count=u& tweets=c` (create **u** fake users and each gets **c** fake tweets)
- `/test/user/<user_id>/tweets?count=t` (generate **t** new fake tweets for this user)
- `/test/user/<user_id>/follow?count=n` (randomly select **n** users to follow this user)