53. Pagination for Posts - Homepage

#pagination #typescript #reactjs #chakraui #frontend

Update the homepage to display paginated posts

- We will use the stack component of chakraui to style the posts - chakra-ui.com/stack
- So we will put a "load more" button at the bottom of the list to trigger loading more posts
- Note that we're using useState() to store the current values for limit and cursor
- The limit value is hardcoded and does not change in this implementation but we could also easily implement a user-selectable limit
- **Note that** this will pull and display the next batch of posts, but **the new batch will replace the previous batch.** We'll fix it in the urql client

/pages/index.tsx

```
import { withUrqlClient } from "next-urql";
import { Layout } from "../components/Layout";
import { usePostsQuery } from "../generated/graphql";
import { createUrqlClient } from "../utils/createUrqlClient";
import NextLink from "next/link";
import {
  Box,
  Button,
  Flex,
  Heading,
  Link,
  Stack,
  Text,
} from "@chakra-ui/react";
import { useState } from "react";

const Index = () => {
  const [postsQueryVariables, setPostsQueryVariables] = useState({
    limit: 10,
    cursor: null as string | null,
  });

  const [{ data, fetching }] = usePostsQuery({
    variables: postsQueryVariables,
});

if (!data && !fetching) {
  return <div>No posts loaded for some reason...</div>;
}

return (
  <>
    <Layout>
      <Flex mb={4} align="center">
        <Heading>LiReddit</Heading>
        <Button ml="auto" type="button" color="teal">
          <Link as={NextLink} href="/create-post">
            Create Post
          </Link>
        </Button>
      </Flex>
      <br />
      {!data ? (
        <div>Loading...</div>
      ) : (
        <Stack spacing={8}>
          {data!.posts.posts.map((p) => (
            <Box key={p.id} p={5} shadow="md" borderWidth="1px">
              <Flex>
                <Heading fontSize="xl">{p.title}</Heading>
                <Flex ml="auto">
                  <Text>posted by:</Text>
                  <Text ml={2} fontWeight="bold">
                    {p.creator.username}
                  </Text>
                </Flex>
              </Flex>
              <Text>{p.text}...</Text>
            </Box>
          ))}
        </Stack>
      )}
      {data && data.posts.hasMore ? (
        <Flex>
```

```
          <Button
            onClick={() =>
              setPostsQueryVariables({
                limit: postsQueryVariables.limit,
                cursor:
                  data.posts.posts[data.posts.posts.length - 1].createdAt,
              })
            }
            isLoading={fetching}
            m="auto"
            my={8}
          >
            Load more...
          </Button>
        </Flex>
      ) : null}
    </Layout>
  </>
 );
};

export default withUrqlClient(createUrqlClient, { ssr: true })(Index);
```

- Note that every time we load the posts, we set the cursor to the createdAt value of the last post in the list, so the next batch starts loading from there:

```
setPostsQueryVariables({
  limit: postsQueryVariables.limit,
  cursor:
    data.posts.posts[data.posts.posts.length - 1].createdAt,
})
```