87. Refresh cache on login/logout

#nextjs #reactjs #frontend

Refresh page on logout

- To refresh the cache on logout we can simply refresh the page with `router.reload()`

/components/NavBar.tsx

```
import { useRouter } from "next/router";

export const NavBar: React.FC<NavBarProps> = ({}) => {
  const router = useRouter();

/* .... */

<Button
  variant="link"
  isLoading={logoutFetching}
  onClick={async () => {
    await logout();
    router.reload();
  }}
>
```

Invalidate posts in the cache on login

- We already are doing this in createPost(). So we extract that logic as a function and use it in both createPost() and login()

/utils/createUrqlClient.ts

```
function invalidateAllPosts(cache: Cache) {
  var previousLimit = cache
    .inspectFields("Query")
    .find((f) => f.fieldName === "posts")?.arguments?.limit as number;
  cache.invalidate("Query", "posts", {
    limit: previousLimit,
  });
}
```

- And then plug it into both createPost() and login()

/utils/createUrqlClient.ts

```
createPost: (result, args, cache, info) => {
  invalidateAllPosts(cache);
},

/* .... */

login: (result, args, cache, info) => {
  // cache.updateQuery({ query: MeDocument }, (data: MeQuery) => { })
  betterUpdateQuery<LoginMutation, MeQuery>(
    cache,
    { query: MeDocument },
    result,
    (r, q) => {
      if (r.login.errors) {
        return q; // return the current query if there's error
      } else {
        return {
          me: r.login.user, // return the user info received from successful
login
        };
      }
    }
  );
  invalidateAllPosts(cache);
},
```