48. Global Error Handling with urql

#reactjs #nextjs #urql #exchange #graphql #frontend #error-handling #authentication

- As it was mentioned in 47. Page - create-post , we could handle the "not authenticated" error, within the create-post page with something like

```
if (error?.message.includes('not authenticated')) {
  router.push("/login");
```

But then we would have to do it like this for every global error (e.g. from the middleware) that is returned from GraphQL. There's a bit more complicated but much better way to do it which we implement through an errorExchange

Implement errorExchange

/utils/createUrqlClient.ts

```
import { pipe, tap } from "wonka";
import router from "next/router";

const errorExchange: Exchange =
  ({ forward }) =>
  (ops$) => {
    return pipe(
      forward(ops$),
      tap(({ error }) => {
        if (error) {
          console.log(error);
          if (error.message.includes("not authenticated")) {
            router.replace("/login");
          }
        }
      })
    );
  };
```

Add errorExchange to the urqlClient

- We insert the errorExchange between cacheExchange and ssrExchange in createUrqlClient(). So the order is as follows:

```
exchanges: [ dedupExchange, cacheExchange, errorExchange, ssrExchange,
fetchExchange, ]
```

Conclusion

- Now this errorExchange catches the global errors and handles them accordingly:

```
if (error.message.includes("not authenticated")) {
            router.replace("/login");
```

- More info at github.com/FormidableLabs/urql/issues/225 - global error handling