


76. Invalidating the cache after deletePost() - Urql Client

[#urql](#) [#cache](#) [#exchange](#) [#mutation](#) [#graphql](#) [#graphcache](#) [#frontend](#)

Add a new update in createUrqlClient for deletePost()

- We use `cache.invalidate()` as we did in [61. Invalidating the cache after createPost\(\) - Urql Client](#) 
- By default, `invalidate()` will make the `post` that we're deleting `null`. So we modify the code in `index.tsx` to prevent trying to read `null` posts

`/utils/createUrqlClient.ts`

```
updates: {  
  Mutation: {  
    deletePost: (result, args, cache, info) => {  
      const { id } = args as DeletePostMutationVariables  
      cache.invalidate({__typename: "Post", id})  
    }  
  },  
}
```

`/pages/index.tsx`

```
return (  
  <>  
    <Layout>  
      {!data ? (  
        <div>Loading...</div>  
      ) : (  
        <Stack spacing={8}>  
          {data!.posts.posts.map((p) =>  
            !p ? null : (  
              <Flex key={p.id} p={5} shadow="md" borderWidth="1px">  
  
                /* ..... */  
              </Flex>  
            )  
          )}  
        </Stack>  
      )  
    </Layout>  
  )  
)
```