CACHING

Currently the most popular and well supported libraries all provide built in caching functionality, further reducing client <-> server requests required. This cache is normalized, allowing even deeply nested objects to be identified and retrieved from cache rather than the database.

CACHING – PART 2

The caching provided brings additional benefits beyond reducing network requests. State management complexity drops drastically because you can simply “re-fetch” the data without actually performing another network request. So rather than the standard flow of:

Request Data -> Set Loading State -> Receive Response -> Render || Error State

We now simply request data, provide a loading and error component up front, and see the end result once the data has been fetched (or retrieved from cache). *The data never has to enter the application’s store*.