

CQS: Command Query Separation

Andreas Wenzelhuemer

October 30, 2022

The topic of this pattern will be about the architectural pattern called Command Query Separation (CQS). CQS splits the backend of an application in two areas, in queries and commands (see figure 1).

- Queries: Are responsible for returning the right values, but never modify anything (there are no side effects).
- Commands: They change the state of the system but do not return any values.

There exist multiple different implementations, but normally there is one object one for representing the operation and a handler class, which contains the actual implementation. A so called dispatcher object is responsible for passing the given request to the correct handler. Additionally here is also an extension of the pattern, the so called Command Query Responsibility Segregation (CQRS). In this pattern, updates and reads are performed on separate databases and the read database gets synchronized when the data in the other database gets updated.

So the paper will contain following:

- CQS: What is it and how to use it.
- CQRS: What's the difference between CQS and CQRS.
- Code examples from a small prototype written in C#

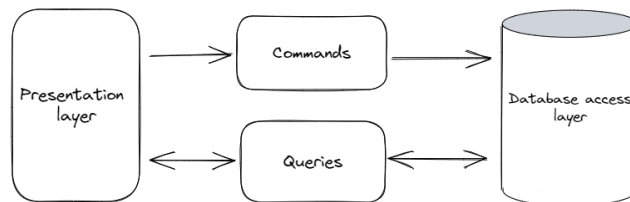


Figure 1: CQS