CQS: Command Query Separation

Andreas Wenzelhuemer

November 3, 2022

The topic of this pattern will be about the architectural pattern called Command Query Separation [1]. CQS is a principle or a guideline, that states that every method should be a command or a query (see figure 1).

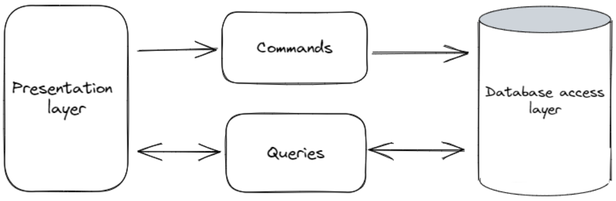

Figure 1: CQS

- Query: Is responsible for returning the correct values without modifying anything (There are no side effects).
- Command: Changes the state of the system but does not return any values.

There exist multiple different implementations, but normally there is one object 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). The difference in contrast to CQS is, that 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 the following topics:

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

[1] `https://martinfowler.com/bliki/CQRS.html`