NCQRS Documentation

Release 0.8

Adam Greene

Nov 07, 2017

Contents

CHAPTER 1

What is NCQRS?

The official answer is:

Ncqrs is a framework for .NET helps build scalable, extensible and maintainable applications by supporting developers apply the Command Query Responsibility Segregation (CQRS) architectural pattern. It does so by providing an infrastructure and implementations for the most important building blocks for command handling, domain modeling, event sourcing, and so. These building blocks help you to focus on the code that adds business value. They come with annotation, convention and configuration support and help you to write isolated and testable.

NCQRS is really a collection of design patterns and technology brought together to allow you to easily and quickly build large scale and scalable applications. They are:

1.1 Command Query Responsibility Segregation

Command Query Responsibility Segregation (CQRS) is a design pattern where, simply put, the parts of your application responsible for updating the state / data of the application is not the same part responsible for querying that state / data. **The Command (write) is separate from the Query (read)**. The reason for doing this is that you scale the write and read portion of your app independently. And if you introduce Event Sourcing and a Message / Event Bus, you can scale both horizontally and vertically.

1.2 Event Sourcing

Event Sourcing uses an append-only store to record the full series of events that describe actions taken on data in a domain, rather than storing just the current state. This store acts as the source of truth or system of record about the current state of the data and can be used to rebuild the read models simply by rerunning the events. This pattern can

simplify tasks in complex domains by avoiding the requirement to synchronize the data model and the business domain; improve performance, scalability, and responsiveness; provide consistency for transactional data; and maintain full audit trails and history that may enable compensating actions.

1.3 Features

- Rich components to support a full CQRS architecture
- Command mapping
- Domain event mapping
- Event sourcing
- Service bus handling of commands and events for microservices architecture
- Event storage with MSSQL, Azure or MongoDB support
- Sample application
- A testing framework to test your events in isolation
- Easy to extend!

CHAPTER 2

Installation