EVENT STORE

Training Course

Advanced Event Sourcing, CQRS and DDD Modelling

Training Course

Advanced Event Sourcing, CQRS and DDD Modelling

Kacper Gunia

This course is delivered by Kacper, an independent software architect, trainer, and consultant, with 6 years of experience of Event Sourcing, CQRS and Domain-Driven Design.

Target audience

Software Developers & Architects with elementary understanding of Event Sourcing

Structure

30% lecture and 70% hands-on coding and exercises

Duration

2 days

Number of attendees

4 - 12 with a single trainer, 13-24 with two

Course Overview

Modelling of a complex IT system is a task that presents many challenges - starting from business requirements, through working with many development teams and ending on non-functional ones related to availability and scalability.

Domain-Driven Design is an approach that focuses on managing this complexity by aligning with the business domains to develop highly maintainable systems that deliver on business requirements.

Two technical patterns often used when implementing DDD are CQRS and Event Sourcing. The Command-Query Responsibility Segregation architecture is a battle-tested approach used to design extremely high scale systems by decoupling reads and writes.

Description

Event Sourcing is an implementation of a persistence model where instead of updating the current state of the system we persist its whole history as a stream of events.

This approach gives us benefits such as a 100% reliable audit log, ability to execute temporal queries, and replays that allow to gain new insights from historical data or correct data inaccurately processed in the past.

In this training you will gain the necessary experience on how to model and implement complex systems using Domain-Driven Design, CQRS and Event Sourcing.

Training Course

Advanced Event Sourcing, CQRS and DDD Modelling

EVENT STORE

Learning Outcomes

- Explore business domain and model it using Design-Level Event Storming and/or Event Modelling
- Model a boundary of an Aggregate based on its business invariants and rules
- Explain the impact of CQRS on Consistency, Availability and Scalability of the system
- Implement Snapshotting of an Aggregate to reduce the time it takes to handle a Command
- Unit test Aggregates and Projections using Given-When-Then formula
- Evaluate event deduplication strategies
- Implement and Replay Projections
- Describe how to scale out an event consumer to meet business SLOs
- Implement a Process Manager supported by a Projection

Scope

DDD building blocks

- Events
- Aggregates
- Commands
- Value Objects
- Entities
- Services

CQRS & Event Sourcing

- Event Sourced persistence Model
- Read and Write stacks of CQRS
- Benefits of Event Sourcing

Aggregate Modelling

- Design-level Event Storming / Event Modelling
- Aggregate Design
- Bounded Contexts
- Private and Public Events

Performance & Scalability

- Eventual consistency
- Monitoring
- Scaling
- Partitioning
- Event based context integration

Aggregate Implementation

- Event Sourced Aggregates
- Task-driven User Interfaces
- Snapshotting
- Testing using Given-When-Then formula
- Correcting Events

Queries & Projections

- Projection design
- Side-effect handling
- Downtime-free Replays
- Process Managers
- Testing using Given-When-Then formula

Prerequisites

- Elementary understanding of Event Sourcing
- Proficiency in one mainstream programming language
- Laptop/notebook with a working development environment and ability to connect to internet
- A sample web application up and running (should be able to accept an http request)
- Docker daemon up and running

Room setup

- Cabaret or boardroom style setup
- Projector / Screen
- Whiteboard / Flipchart

EVENT STORE®