

## Training is withdrawn

Training program:

### **Command-query Responsibility Segregation architecture and Event Sourcing persistence**

Info:

<b>Name:</b>	<b>Command-query Responsibility Segregation architecture and Event Sourcing persistence</b>
<b>Code:</b>	<b>ddd-CqRS</b>
<b>Category:</b>	Domain Driven Design and Event Storming
<b>Target audience:</b>	architects developers devops
<b>Duration:</b>	2 days
<b>Format:</b>	50% lecture / 50% workshop

---

Command Query Responsibility Segregation and Event Sourcing Training shows alternative ways to manage commands and queries in your application.

This training is dedicated to advanced programmers, designers and architects who look for solving the complexity of business problem and high load.

During the workshop we leverage Event Storming and tactical/strategic modeling taken from Domain-Driven Design.

# Training program

## 1. Motivation

1.1. Classical way of storing state and its limitations

1.2. Audit

1.3. Temporal queries

1.4. Single Source of Truth

1.5. What is a domain event?

1.6. Big Picture Event Storming

## 2. Modeling

2.1. Understand Bounded Contexts and Aggregates

2.2. Command vs Query

2.3. Design Level Event Storming

2.4. Aggregates backed by Event Sourcing - best practices

2.5. Unit Testing of Event Sourced Aggregates

2.6. Event Sourcing as Functional Programming

2.7. Aggregates which manage state and emit events

## 3. Persistence

3.1. Application layer (use-cases)

3.2. Introduction to Hexagonal Architecture

3.3. Compare market tools

3.4. Event Store as a database and message broker

3.5. Scalable Atom Feed

3.6. Partitioning

## 4. Read your data

4.1. Snapshots and Projections

4.2. Optimal storage for queries

4.3. CQRS

4.4. Parallel Models

## 5. Microservices and CQRS/ES

5.1. Eventual Consistency

5.2. Horizontal Asymmetric Scaling

5.3. Blue/Green Deployments

5.4. API Gateway

5.5. Aggregate Services

## 6. Advanced Problems

6.1. Versioning your events

6.2. Read your own writes

6.3. Monotonic reads

6.4. Retrospective events

6.5. Compensation events

6.6. Communication between writes and reads

6.7. GDPR