BO·TT·EGA
IT minds

Program szkolenia:

Architektura Command-query Responsibility Segregation

Informacje:

Nazwa:	**Architektura Command-query Responsibility Segregation**
Kod:	**DDD-CqRS**
Kategoria:	Domain Driven Design
Grupa docelowa:	architekci developerzy
Czas trwania:	1 dzień
Forma:	70% wykłady / 30% dyskusja

Szkolenie prezentuje architekturę CqRS wraz z alternatywnymi podejściami do implementacji stosu Read i Write.

Szkolenie jest przeznaczone dla zaawansowanych programistów, projektantów i architektów poszukujących rozwiązań dla systemów o złożonej logice i pracujących pod dużym obciążeniem

Z uwagi na specyfikę tematu podczas szkolenia nie przeprowadzamy warsztatów praktycznych (gdyż implementacja mocno zależy od kontekstu technicznego), zamiast tego oferujemy dyskusję nad stosowalnością podejścia w kontekście problemów z jakimi pracują uczestnicy.

Zalety szkolenia:

- Nowoczesne architektura (wspierająca DDD)
- Podejście do skalowania systemu

BOTTEGA IT minds

Szczegółowy program:

- **1. Ogólna idea**
  - 1.1. paradygmat Command-query Separation
  - 1.2. Problemy z podejściem warstwowym
    - 1.2.1. Wydajność maperów relacyjno-obiektowych
    - 1.2.2. Stosowalność baz relacyjnych
  - 1.3. Rozszerzenie architektury warstwowej do dwóch stosów
- **2. Trzy podejścia do separacji Query**
  - 2.1. Odpytywanie wprost modelu domenowego
    - 2.1.1. Zwracanie DTO, jak robić to w Hibernate
    - 2.1.2. Problemy z wydajnością ORM do tego typu zastosowań
    - 2.1.3. Praktyczne przykłady wykorzystania np MyBatis
  - 2.2. Widoki zmaterializowane
  - 2.3. Zdarzenia emitowane z domeny odświeżające model do odczytu
    - 2.3.1. Zdarzenia z modelu domenowym
    - 2.3.2. Semantyka zdarzeń
    - 2.3.3. Pułapki Event Driven Architecture
      - 2.3.3.1. Single Point of Failure
      - 2.3.3.2. Eventual Consistency
    - 2.3.4. Wykorzystanie noSQL w ReadModel
- **3. Implementacja stosu Write**
  - 3.1. Wzorzec Command i CommandHandler - implementacja
    - 3.1.1. Dekorowanie logiki
    - 3.1.2. Odporność na "biznesowe" ataki DOS

BO·TT·EGA
IT minds

3.1.3. Systemy multi-tenant

3.2. Rozwarstwienie logiki

3.2.1. Logika Aplikacyjna - model Use Case/User Story

3.2.2. Logika Domenowa - DDD

4. Event Sourcing - jako jedno z podejść do persystencji domeny

4.1. Idea i szablon implementacji

4.2. Klasy problemów jakie można w ten sposób rozwiązać

4.3. Kiedy NIE stosować ES

5. Architektura Ports and Adapters (Hexagonalna) jako spójny model architektoniczny