

Program szkolenia:

Architektura Command-query Responsibility Segregation oraz Event Sourcing

Informacje:

| | |
|----------------------|--|
| Nazwa: | Architektura Command-query Responsibility Segregation oraz Event Sourcing |
| Kod: | DDD-CqRS |
| Kategoria: | Domain Driven Design i Event Storming |
| Odbiorcy: | DevOps, developerzy, architekci |
| Czas trwania: | 2 dni |
| Forma: | 50% wykłady / 50% warsztaty |

Szkolenie prezentuje architekturę CqRS wraz z alternatywnymi podejściami do implementacji stosu Read i Write. Event Sourcing jest jedną z opcji przechowywania danych w stosie Write.

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

W trakcie szkolenia będziemy podejmować decyzje architektoniczne bazując na modelowaniu taktycznym i strategicznym DDD stosując Event Storming.

Z uwagi na specyfikę techniczną podczas szkolenia możemy wybrać stos technologiczny (Java, .Net, PHP, Ruby, Node.js).

Zalety szkolenia:

- Elementy DDD i Event Stormingu
- Kontekst Microservices i Cloud
- Kilka podejść do integracji komponentów architektury (np. bez Event Sourcingu)

Szczegółowy program:

1. Motywacja Event Sourcingu

1.1. Problemy z klasycznym przechowywaniem stanu

1.2. Audyt

1.3. Temporal queries

1.4. Single Source of Truth

1.5. Czym jest zdarzenie domenowe?

1.6. Big Picture Event Storming

2. Modelowanie

2.1. Zrozumienie konceptu Bounded Contextów oraz Agregatów

2.2. Różnica pomiędzy komendą, a zdarzeniem

2.3. Design Level Event Storming

2.4. Modelowanie agregatów w ujęciu Event Sourcing - najlepsze praktyki

2.5. Testowanie jednostkowe agregatów w ujęciu Event Sourcing

2.6. Event Sourcing jako programowanie funkcyjne

2.7. Agregaty, które zmieniają stan i produkują zdarzenia

3. Persystencja

3.1. Serwisy aplikacyjne

3.2. Wprowadzenie do architektury hexagonalnej

3.3. Porównanie dostępnych narzędzi

3.4. Event store jako baza danych i broker wiadomości

3.5. Skalowalny Atom Feed

3.6. Skalowalność poprzez partycjonowanie

4. Odczyt danych

4.1. Snapshoty i Projekcje

4.2. Optymalizacja modelu do odczytu

4.3. CQRS

4.4. Parallel Models

5. Świat microserwisow i CQRS/ES

5.1. Eventual Consistency

5.2. Horyzontalne asymetryczne skalowanie

5.3. Blue/Green Deployments

5.4. Dopasowany model danych

5.5. API Gateway

5.6. Serwis agregujący zdarzenia ze strumieni z różnych mikroserwisów

6. Zaawansowane problemy

6.1. Wersjonowanie schematu zdarzeń

6.2. Problem "Read your own writes"

6.3. Problem monotonicznych odczytów

6.4. Retrospective events

6.5. Compensation events

6.6. Komunikacja pomiędzy modelem do odczytu i zapisu