

## Program szkolenia:

# noSQL - architektura i przegląd możliwości w ujęciu CqRS

## Informacje:

<b>Nazwa:</b>	<b>noSQL - architektura i przegląd możliwości w ujęciu CqRS</b>
<b>Kod:</b>	<b>nosql-rev</b>
<b>Kategoria:</b>	NoSQL
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

Celem szkolenia jest świadomy wybór rozwiązań składowania danych alternatywnych w stosunku do SQL. Przegląd i analiza różnych kategorii baz danych oraz skorelowanie ich z klasycznymi bazami relacyjnymi pozwoli na szybsze przyswojenie tematu, dla uczestników zaznajomionych z bazami SQL.

W trakcie szkolenia uczestnicy zdobędą umiejętność wyboru kategorii rozwiązania składowania danych i poznają różne sposoby ich modelowania, dostosowane do wymagań biznesowych oraz wydajności. Nie zabranie w związku z tym wskazówek odnośnie doboru odpowiedniej architektury, w tym także modeli hybrydowych, złożonych zarówno z baz SQL oraz NoSQL.

Szkolenie adresuje również metody radzenia sobie z ograniczoną spójnością danych, wprowadzając w świat DDD oraz CQRS. Podejście polyglot persistence będzie przedstawione na przykładzie architektury Command-query Responsibility Segregation gdzie model domenowy może być przechowywany np w bazie dokumentowej a wydajny model do odczytu np w bazie grafowej.

Zapoznanie się z szeroką gamą dostępnych rozwiązań oraz stosowanymi modelami pozwala na podjęcie świadomej i lepszej decyzji na poziomie architektury systemu.

Największym atutem szkolenia jest część warsztatowa, która umożliwi zweryfikowanie teorii w praktyce. Uczestnicy uczą się również metod integracji nowo poznanych rozwiązań z rozwijanymi systemami, co pozwala w sposób efektywny wykorzystywać wiedzę nabytą w trakcie szkolenia, bezpośrednio po jego zakończeniu.

## Zalety szkolenia:

- realna złożoność przykładów
- najlepsze praktyki
- dobór klasy rozwiązania do klasy problemu

## Szczegółowy program:

### 1. Typy baz danych - dobór do klasy problemu

- 1.1. Bazy typu klucz-wartość
- 1.2. Bazy kolumnowe
- 1.3. Bazy dokumentowe
- 1.4. Bazy grafowe
- 1.5. Bazy szeregów czasowych (time-series)

### 2. Porównanie baz NoSQL z relacyjnymi bazami SQL

- 2.1. ACID vs. CAP i PACELC
- 2.2. Skalowalność rozwiązań
- 2.3. Wyzwania wynikające ze stosowania baz rozproszonych

### 3. Modelowanie danych - dobór do klasy problemu

- 3.1. Płaskie agregaty
- 3.2. Zagnieżdżone obiekty
- 3.3. Miękkie relacje

### 4. Spójność danych - podejście strategiczne

- 4.1. Eventual consistency a spójność wg ACID
- 4.2. Projektowanie systemów eventual consistency
- 4.3. W jaki sposób ograniczona spójność pomaga w zrozumieniu domeny?

### 5. Dostęp do baz NoSQL z poziomu aplikacji

- 5.1. Dostęp za pośrednictwem dedykowanych sterowników
- 5.2. Wykorzystanie Spring Data jako abstrakcji dostępu do danych

### 6. Przygotowanie systemu, wykorzystującego różne typy baz danych (ilość wykorzystanych rozwiązań dostosowana do ilości czasu i potrzeb)

- 6.1. Redis jako rozwiązanie typu klucz-wartość

6.2. Apache Cassandra jako baza kolumnowa

6.3. Elasticsearch lub MongoDB jako baza dokumentowa

6.4. Neo4j lub OrientDB jako przykład wykorzystania bazy grafowej

6.5. Elasticsearch lub InfluxDB jako baza szeregów czasowych

## 7. Architektura Command-query Responsibility Segregation

7.1. Write model - dobór bazy do natury domeny

7.2. Read model - dobór bazy do natury zapytań

7.3. Integracja modeli