BOTTEGA IT minds

Program szkolenia:

Nowoczesna architektura aplikacji web - Microservices, REST, noSQL (Java/.NET)

Informacje:

Nazwa:	**Nowoczesna architektura aplikacji web - Microservices, REST, noSQL (Java/.NET)**
Kod:	**arch-ms-workshop-modern**
Kategoria:	Warsztaty eksperckie Microservices
Grupa docelowa:	architekci developerzy
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Szkolenie przeznaczone dla projektantów i architektów, którzy chcą odnaleźć się w świecie nowych technologii tworzenia lekkich a zarazem ultra wydajnych i skalowalnych aplikacji webowych.

Zalety szkolenia:

- Nowoczesne architektury (CqRS - wspierająca DDD)
- Integracja technologii i całościowe podejście
- Dobór klasy rozwiązania do klasy problemu

BOTTEGA IT minds

Szczegółowy program:

1. REST

- 1.1. Dobrze zrozumieć ważne aspekty http
- 1.2. Filozofia REST API
  - 1.2.1. Standardy
  - 1.2.2. Najlepsze praktyki - dobór do kontekstu
- 1.3. Poprawna interpretacja SOA
  - 1.3.1. Model kanoniczny a domenowy
- 1.4. Cel: dobrze zaprojektowany rest pokrywa jednocześnie zarówno aplikację webową jak i technologie mobilne

2. Microservices i architektura CqRS

- 2.1. Założenia
  - 2.1.1. Projektowanie z myślą o awariach
  - 2.1.2. Podejście ewolucyjne
  - 2.1.3. Decentralizacja zarządzania danymi
  - 2.1.4. Endpoint a Pipe
- 2.2. Strategie refaktoryzacji systemów monolitycznych
- 2.3. Jak określić granicę serwisów
  - 2.3.1. Podejście Bounded Context z DDD
  - 2.3.2. Antywzorzece: Nanoservice
- 2.4. Testowanie akceptacyjne serwisów

3. Architektura systemu

- 3.1. Integracja serwisów
- 3.2. Command-query Responsibility Segregation
  - 3.2.1. Stos write - model domenowy

BOTTEGA IT minds

- 3.2.1.1. Building Blocks DDD
- 3.2.1.2. Testowanie jednostkowe logiki domenowej
- 3.2.1.3. Event Sourcing
- 3.2.2. Stos read - denormalizacja w celu optymalizacji odczytów

4. Bezpieczeństwo

- 4.1. SSO
- 4.2. Ataki i zabezpieczenia przed nimi
  - 4.2.1. XSS, XML External Entity, XML Entity Expansion i Session Fixation

5. Skalowalne systemy rozproszone

- 5.1. Zdarzenia domenowe
- 5.2. Orkiestracja zdarzeń, model Sagi
- 5.3. Kolejki
  - 5.3.1. Wybór rozwiązania
  - 5.3.2. Optymalizacja
  - 5.3.3. Wzorce
    - 5.3.3.1. Event Broker
    - 5.3.3.2. Event Bus
- 5.4. Podejście do Eventual Consistency

6. Architektura aplikacji klienckicj opartych o Angular.js

- 6.1. 2 style
  - 6.1.1. Single Page Apps (z odmianami MV* - MVP, MVVM itd.),
  - 6.1.2. ROCA style (Resource Oriented Client Architecture)
- 6.2. Angular.js
  - 6.2.1. Wprowadzenie do developmentu aplikacji webowych
- 6.3. Wyprostowanie błędów w programowaniu w JS

BOTTEGA IT minds

6.4. Testy, zarządzanie zależnościami

7. noSQL

7.1. Kiedy warto i do czego

7.2. MongoDB/RavenDB

7.3. CAP theorem w praktyce

7.4. Projektowanie modeli pod kątem odczytu

8. Continuous Integration i Continuous Deployment

8.1. Umożliwianie CD ze strony architektury i kodu

8.2. Narzędzia