

## Program szkolenia:

# Microservices - podejście kompleksowe oparte o DDD i Cloud

## Informacje:

<b>Nazwa:</b>	<b>Microservices - podejście kompleksowe oparte o DDD i Cloud</b>
<b>Kod:</b>	<b>DDD-complex</b>
<b>Kategoria:</b>	Domain Driven Design i Event Storming
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	

---

Szkolenie jest przeznaczone dla osób, które chcą rozwinąć swoje myślenie o architekturze aplikacji opartych o Spring Boot.

Na poziomie architektury wdrożeniowej zastosujemy podejście microservices z wykorzystaniem Spring Cloud

Na poziomie architektury systemowej podejmiemy złożony problem biznesowy wymagający strategicznego podejścia do modelowania (DDD) a co za tym idzie podziału na moduły oraz racjonalnego podejścia do ich integracji. Przejdziemy przez kilka scenariuszy integracyjnych oraz kilka technik ich implementacji.

Na poziomie architektury aplikacji posłużymy się nowoczesnymi wzorcami i stylami architektonicznymi.

Na koniec nasz system nie będzie przypominał hello world z tutoriali, zostaniemy z recepturą postępowania odpowiednią do nietrywialnych projektów.

## Zalety szkolenia:

- Zintegrowane podejście łączące analizę, architekturę i framework Spring
- Nietrywialne przykłady modelowania
- Dbanie o każdy poziom architektury

## Szczegółowy program:

### 1. Architektura w stylu C4 - projekt systemu warsztatowego

- 1.1. Context - otoczenie biznesowe
- 1.2. Containers - arch. wdrożeniowa
- 1.3. Components - arch systemu
  - 1.3.1. Integracja modułów
  - 1.3.2. Wzorce strategiczne
- 1.4. Classes - arch. aplikacji, wzorce taktyczne

### 2. Dekompozycja funkcjonalna

- 2.1. Domain-Driven Design na poziomie strategicznym
- 2.2. Event Storming
- 2.3. Motywacja, Conway's Law
- 2.4. Decentralizacja danych / Polyglot Persistence
- 2.5. Strategie określania granic serwisu
- 2.6. Problem Single Point of Failure
- 2.7. Typowe pułapki przy rozbijaniu monolitu

### 3. Spring i Spring Boot - szybkie prowadzenie

- 3.1. Motywacja
- 3.2. Proxy
- 3.3. Spring Bean
- 3.4. Podstawowe elementy autokonfiguracji

### 4. REST ze Spring MVC

- 4.1. HTTP jako protokół aplikacyjny
- 4.2. Zalety i pułapki cache

4.3. Zasoby i ich reprezentacje

4.4. Poziomy dojrzałości REST

4.5. REST jako obieg dokumentów, a warstwa domenowa serwisu

4.6. HATEOAS jako mechanizm maszyny stanów

4.7. Wersjonowanie API

4.8. Typowe pułapki

## 5. Architektura serwisu

5.1. Model domeny i model danych

5.2. Spring Data JPA and Spring Data MongoDB

5.3. Granice obiektów

5.4. Test-Driven Development i reguł biznesowych

5.5. Onion Architecture / Ports and Adapters

5.6. Warstwa aplikacyjna jako sterownik procesu

5.7. Eventual consistency

5.8. Command Query Responsibility Segregation

5.9. Zdarzenia domenowe ze Spring Application Events

## 6. Integracja systemów rozproszonych (Spring Cloud Netflix)

6.1. Service Discovery

6.2. Client-Side Load Balancing

6.3. Hystrix jako Cirtcut Breaker

6.4. Distributed Tracing z Zipkinem

6.5. Config Server

6.6. Monitorowanie

## 7. Integracja systemów rozproszonych (Spring Cloud Stream)

7.1. Event vs Command vs Message

7.2. Process Manager/Saga

7.3. Kolejki

7.4. Event Broker i Event Store

7.5. Skalowalny Atom Feed

7.6. Rozszerzalność/pluginowość architektury