

## Program szkolenia:

# Usługi w modelu RESTful

### Informacje:

<b>Nazwa:</b>	<b>Usługi w modelu RESTful</b>
<b>Kod:</b>	<b>java-arch-REST</b>
<b>Kategoria:</b>	Architektura Java
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Szkolenie przedstawia rzetelne podstawy projektowania i tworzenia usług sieciowych w stylu RESTful. Szkolenie może zostać dopasowane do kompleksowego rozwiązania osadzonego w Java EE jako moduł szkoleniowy służący do udostępniania funkcjonalności opracowanych podczas szkolenia EJB lub Spring.

### Zalety szkolenia:

- Poruszamy zagadnienia zaawansowane
- Najlepsze praktyki i wzorce
- Projektowanie usług zgodnie z kanonami sztuki

## Szczegółowy program:

### 1. Projektowanie Usług Sieciowych

1.1. Aspekty ziarnistości

1.2. Aspekty Wydajności

1.3. Typowe błędy w projektowaniu API

1.4. Wykorzystanie REST w architekturze mikro-serwisów

1.4.1. Wykorzystanie Spring Boot w architekturze mikro-serwisów

1.5. Komunikacja synchroniczna i asynchroniczna

### 2. Przegląd i porównanie podejścia JAX-WS i JAX-RS

2.1. Protokół SOAP vs HTTP

2.1.1. Operacje vs zasoby

2.1.2. Mapowanie błędów

2.1.3. WSDL

2.1.4. WADL

2.2. Porównanie zastosowań

2.2.1. Zalety i ograniczenia restrykcyjnego kontraktu

2.2.2. REST sposobem na ograniczenie boilerplate kodu

### 3. Dojrzałość API

3.1. XML / JSON

3.2. Zasoby

3.3. Czasowniki HTTP

3.4. Negocjacja zawartości

### 4. Wzorce wykorzystywane w usługach RESTowych

4.1. Wybór odpowiedniej ścieżki dla zasobu

4.2. Mapowania operacji HTTP na operacje na zasobach

4.3. Reprezentacja zasobów

4.3.1. Oróżnienie zasobu od jego formy i wersji

4.4. Idempotentność

4.5. Kody błędów

4.6. Uwierzytelnianie

4.7. Wersjonowanie API

4.7.1. Typowe błędy

4.7.2. Najlepsze praktyki

4.8. HATEOAS

4.9. Stronnicowanie zasobów

4.10. Cache - pułapki i najlepsze praktyki

## 5. Deployment aplikacji wystawiającej usługę RESTową na serwerze EE

5.1. Konfiguracja projektu

5.2. Deployment na wybranym serwerze

## 6. Testowanie

6.1. Komunikacja z serwerem

6.2. Scenariusze testowe

6.2.1. Weryfikacje

6.2.2. Mocki

6.2.3. Skrypty

6.3. Testy wydajnościowe

6.4. Testy bezpieczeństwa

6.5. Pułapki stosowania SoapUI w automatycznych testach

6.5.1. Przygotowanie danych

6.5.2. Zależności biblioteczne

6.5.3. Kontrola wersji

6.6. Testy kontraktu API

6.7. Alternatywne podejścia

## 7. Budowanie usług za pomocą JAX-RS lub Spring MVC

7.1. Mapowanie zasobów

7.1.1. Ścieżki

7.1.2. Operacje

7.1.3. Produces/Consumes

7.1.4. Sposoby przekazywania parametrów

7.2. Mapowanie komunikatów

7.2.1. Contract Last

7.2.2. Contract First

## 8. Budowa aplikacji klienckich

8.1. Retrofit

8.2. Spring RestTemplate

8.3. Jersey / REStEasy

## 9. Zagadnienia zaawansowane

9.1. Skalowalność

9.2. Bezpieczeństwo

9.2.1. Oauth 2

9.2.2. JWT