

## Program szkolenia:

# Java Server Faces 2

### Informacje:

|                        |                             |
|------------------------|-----------------------------|
| <b>Nazwa:</b>          | <b>Java Server Faces 2</b>  |
| <b>Kod:</b>            | <b>Java-EE-JSF 2</b>        |
| <b>Kategoria:</b>      | Java EE                     |
| <b>Grupa docelowa:</b> | developerzy                 |
| <b>Czas trwania:</b>   | 3 dni                       |
| <b>Forma:</b>          | 50% wykłady / 50% warsztaty |

---

Program został skontrowany tak aby przedstawić zagadnienia w kontekście konkretnych problemów, które ilustrują praktyczne wykorzystanie każdej z funkcjonalności JSF.

Program szkolenia zawiera rozszerzenie o najlepsze praktyki projektowe i architektoniczne.

Tematyka szkolenia obejmuje wszystkie cechy frameworka JSF 2.0 istotne z punktu widzenia developera aplikacji.

### Zalety szkolenia:

- Prezentujemy typowe jak i mniej typowe pułapki
- Przykłady o realnym poziomie komplikacji
- Duży zakres przydatnych trików i sztuczek

## Szczegółowy program:

### 1. Wstęp do Servlet API

1.1. Servlety, Filtry, Zdarzenia kontenera

### 2. Architektura aplikacji web

2.1. Warstwy i ich charakterystyka

2.2. Przegląd podejść warstwowych

2.2.1. Miejsce Encji JPA w modelu

2.3. Ewolucja MVC

2.3.1. MVP, MVVC

### 3. Ogólna idea frameworka

3.1. Klasy systemów do których się aplikuje oraz przypadki gdy NIE jest odpowiedni

### 4. JSF 1.2 – krótki wstęp i wprowadzenie w kontekst nowej wersji

4.1. Typowe problemy i ich obejścia

4.2. Przegląd istniejących rozwiązań, które stały się inspiracją dla JSF 2.0

### 5. Konfiguracja i przegląd narzędzi

### 6. Wstęp do Facelets

6.1. Szablony

6.1.1. Parametry szablonów

6.1.2. Wylizanie szablonów w runtime

6.1.3. Technika reużywalnych "płytek"

6.2. Różnice w stosunku do JSP

6.3. Konwencje i najlepsze praktyki

### 7. Cykl życia żądania – aspekty praktyczne

### 8. Model komponentów

8.1. Wygodne adnotacje dla backing beanów i komponentów pomocniczych

8.2. Wykorzystanie Standardowych adnotacji JSR-250

8.3. Zasięgi komponentów i ich zastosowanie – dobór do problemu

8.3.1. Zasięg View - rozwiązanie problemu klikalnych list

8.3.2. Zasięg Flash w kontekście paradygmatu POST GET REDIRECT

8.4. Inicjowanie modelu w Stylu Pull

8.5. Metody callback – nieocenione narzędzie

8.6. Wstrzykiwanie zależności

8.6.1. Techniki separacji modeli widoku i kontrolerów

8.6.2. Techniki optymalizacji zużycia pamięci dzięki ograniczaniu czasu życia obiektów

## 9. Walidacja modelu

9.1. Walidatory Standardowe

9.2. Tworzenie własnych walidatory

9.3. Składowe standardu The Bean Validation JSR (JSR-303)

9.3.1. Wsparcie ze strony JSF

## 10. Konwersja obiektów

10.1. Konwertery standardowe (automatyczne) i własne

10.2. Typowe zastosowania: listy rozwijane o wartościach obiektowych

## 11. Komponenty wizualne

11.1. Bindowanie

11.2. Manipulacja

11.3. Projektowanie formularzy pod kątem testowalności Selenium

11.3.1. Wykorzystanie CSS do nadawania semantycznej struktury stronom

## 12. Wsparcie dla REST

12.1. Problemy z JSF 1.x

12.2. Styl Pull (napełnianie modelu przy pomocy reguł nawigacji lub adnotacji)

12.3. Nawigacja zorientowana na URL i zagadnienie bookmarkable

### 13. Model nawigacji

13.1. Reguły warunkowe

13.2. Uproszczenie mapowania

13.3. Wsparcie dla GET

13.4. Obsługa błędów

13.5. Orientacja nawigacji na RESTfull

13.6. Najlepsze praktyki odnośnie wyboru stylu nawigacji

### 14. Wygodne tworzenie własnych komponentów

14.1. Wykorzystanie Facelets - komponenty kompozytowe

14.2. Standardowa konwencja

14.3. Zarządzanie zasobami komponentów

### 15. Wsparcie dla AJAX

15.1. Zasada działania

15.2. API dla AJAX

15.3. Techniki optymalizacji działania formularzy

### 16. Koncepcja Etapów/Poziomów Projektu oraz jej praktyczne wykorzystanie

### 17. Typowe zagadnienia i problemy

17.1. Formularze master-detail

17.2. Formularze edycja/podgląd

17.3. Listy i konwertery obiektów

17.4. Wyskakujące okienka

### 18. Testowanie warstwy prezentacji

18.1. Behaviour Driven Development

18.2. Techniki Scenariuszy akceptacyjnych

18.3. Narzędzia

18.3.1. JBehave - wykonywalne specyfikacje

18.3.2. Selenium - automatyczna interakcja jako implementacja scenariuszy w JBehave