

Program szkolenia:

Aplikacje Web oparte o Seam Framework

Informacje ogólne

Nazwa:	Aplikacje Web oparte o Seam Framework
Kod:	Seam
Kategoria:	Seam
Grupa docelowa:	projektanci, architekci, programiści
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Prezentacja kompleksowego zestawu rozwiązań pozwalających na produktywnie tworzenie profesjonalnych aplikacji webowych z wykorzystaniem technologii JEE zintegrowanych przy użyciu Seam Framework.

Koncepcja szkolenia została oparta o zestaw praktycznych problemów jakie stają przed developerem – prezentujemy sprawdzone i najlepsze ich rozwiązania w Seam.

Techniki Domain Driven Design zostały dopasowane do realiów rzeczywistych projektów.

Szkolenie swym zakresem wychodzi daleko poza ramy typowych, szablonowych przewodników oraz innych szkoleń dostępnych na rynku.

Podczas wykładów poruszamy zaawansowane zagadnienia projektowe oraz omawiamy uwarunkowania i konteksty, w których warto korzystać z pewnych zaawansowanych mechanizmów Seam.

Podczas warsztatów rozwiązujemy problemy w realistyczny sposób, tak jak wymagają tego standardy w realnych projektach. Stosowane podczas warsztatów techniki wykraczają poza proste naiwne rozwiązania i stosujemy podejścia zapewniające większą wydajność oraz ergonomię użytkownika: zaawansowane wykorzystanie AJAX, integracja z jQuery architektura oparta o zdarzenia, przetwarzanie asynchroniczne.

W trakcie szkolenia wskazujemy typowe pułapki, często popełniane błędy oraz sprawdzone sposoby obejścia problemów.

W trakcie ćwiczeń praktycznych budowana jest spójna aplikacja web, której wymagania funkcjonalne przekładane są na kolejno prezentowane konstrukcje frameworka.

Zalety szkolenia:

- » Dobór architektury aplikacji
- » Aspekty wydajności
- » Mocno rozszerzony program

Program szkolenia:

1. Konfiguracja standardowego projektu EJB lub lekkiego projektu POJOs

1.1. Konfiguracja produktywnego środowiska developerskiego (WTP Eclipse Plugin i JBoss Eclipse Plugin)

1.2. Maven

1.3. Wsparcie narzędzi dla Test Driven Development

1.4. Lekkie systemy webowe oparte o serwer Tomcat (na życzenie)

2. Dobór architektury

2.1. Standardowa architektura aplikacji Seam - wady i zalety

2.2. Architektoniczne aspekty reużywalności kodu i jego podatności na testy

2.3. Architektura warstwowa dla bardziej zaawansowanych projektów (na życzenie)

2.3.1. Wzorce architektoniczne

2.3.2. Wzorce projektowe stosowane w aplikacjach webowych

2.3.3. Wzorce porządkowania logiki biznesowej

2.4. Architektura warstwowa dla podejścia Domain Driven Design – oparte o unikatowe prace badawcze Bottega-Pracownia Programistyczna (na życzenie)

2.5. Architektura wspierająca Restfull i SOA

2.6. WebServices i Przygotowanie architektury zorientowanej na usługi i wymianę wspólnych danych

3. Komponenty

3.1. Zasięg komponentów (kontekst) i jego dobór do problemu

3.2. Podejście stanowe vs bezstanowe - zastosowania

3.3. Cykl życia komponentu - pułapki

3.4. Przydatne komponenty wbudowane

3.5. Sposoby dostępu do komponentów i komunikacja międzykomponentowa

3.6. Zależności w Seam: bijecka zależności, produktywne techniki

3.7. Praktyczne zastosowanie interceptorów (wprowadzenie technik Aspect Oriented oraz budowa własnych frameletów)

4. Praktyczne zastosowania konfiguracji przez adnotacje i przez XML

5. Konwersacje oraz ich pułapki

5.1. Kontekst konwersacji

5.2. Granice konwersacji

5.3. Łączenie i przełączanie konwersacji

6. Modele nawigacji i ich dobór do problemu

6.1. Nawigacja zorientowana na strony

6.2. Nawigacja zorientowana na zdarzenia

6.3. Styl pull vs push

6.4. interceptory

6.5. restfull

7. Walidacja

7.1. automatyczna

7.2. własna

7.3. obsługa błędów

8. Dostęp do danych

8.1. Transakcje (konfiguracja, poziomy izolacji)

8.2. Zarządzanie persystencją

8.3. Konwersacje (rozszerzony tryb persystencji)

9. Warstwa prezentacji

9.1. JSF

9.2. Facelets

9.3. RichFaces

9.4. AJAX i Seam Remoting - integracja z jQuery

9.5. Standardowe problemy: Master-detail, listy, Back-button,...

10. Mailing

10.1. Zaawansowana architektura podsystemu mailowego

10.2. Techniki asynchronicznego wysyłania maili

11. Wewnętrzny framework (Seam application framework)

11.1. zastosowania

11.2. pułapki

11.3. produktywne prototypowanie aplikacji

12. Bezpieczeństwo

12.1. Autentykacja i autoryzacja

12.2. Funkcja „zapamiętaj mnie”

12.3. Powrót na żądane strony po zalogowaniu

13. Standardowe problemy

13.1. Master-detail

13.2. Dekorowanie wierszy tabel i prezentacja dodatkowych danych

13.3. Komboboksy i konwertery

13.4. Wyskakujące okienka (komponenty i AJAX)

13.5. Wycieki pamięci

13.6. Servlety (zanurzenie i dostęp do kontekstu Seam)

13.7. Dostęp do danych (różne style i ich zastosowanie)

14. Zagadnienia zaawansowane:

14.1. Optymalizacja i tuning Seam

14.2. Modułowa architektura oparta o komponenty sterowane systemem zdarzeń Seam

14.3. Techniki optymalizacji warstwy dostępu do danych (JPA/Hibernate) oraz unikanie pułapek wydajnościowych:

14.3.1. cache zapytań

14.3.2. cache encji

14.3.3. wydajność zapytań

14.3.3.1. n+1Select problem - wykrywanie i zapobieganie

14.3.4. projektowanie wydajnego modelu

14.4. Techniki tworzenia reużywalnych szablonów widoku w Facelets

14.5. Tworzenie własnych komponentów JSF

14.6. Techniki optymalizacja warstwy prezentacji – redukcja narzutów RichFaces

14.7. Integracja (do wyboru na życzenie)

14.7.1. Hibernate - korszytanie wprost z API Hibernate, konfiguracja cache

14.7.2. jBPM - procesy biznesowe i flow stron (integracja z komponentami Seam)

14.7.3. Drools

14.7.4. iText - generowanie PDF na podstawie komponentów Seam

14.7.5. BIRT - generowanie raportów z wykorzystaniem stanu komponentów Seam

15. Narzędzia

15.1. jBoss Tools