BO·TT·EGA
IT minds

Program szkolenia:

Tworzenie aplikacji w Ruby on Rails z wykorzystaniem zwinnych metodyk

Informacje:

Nazwa:	**Tworzenie aplikacji w Ruby on Rails z wykorzystaniem zwinnych metodyk**
Kod:	**RoR-Rails**
Kategoria:	Ruby on Rails
Grupa docelowa:	developerzy
Czas trwania:	4 dni
Forma:	30% wykłady / 70% warsztaty

Podczas tego szkolenia uczestnicy dowiedzą się, w jaki sposób szybko tworzyć wysokiej jakości aplikacje webowe w oparciu o technologię Ruby on Rails.
Poznają oni kluczowe możliwości najnowszej wersji Rails - 3.2, jak również elementy języka Ruby w wersji 1.9 i nauczą się, jak wykorzystać tę wiedzę do tworzenia aplikacji, krok po kroku - od idei do wdrożenia.

W oparciu o tworzoną od podstaw aplikację uczestnicy poznają najważniejsze cechy tej technologii.

Szkolenie ma formę wykładów zakończonych warsztatami, główny nacisk położony jest na ćwiczenia praktyczne.
W procesie tworzenia projektu przedstawione zostaną elementy zwinnych (agile) metod tworzenia oprogramowania oraz praktyk takich jak refaktoring, DRY i innych.

Zalety szkolenia:

- Nauka poprzez ćwiczenia praktyczne
- Poznanie pełnego procesu wytwarzania oprogramowania w oparciu o metodyki zwinne (TDD, BDD, automatyzacja)
- Zawiera elementy Domain Driven Design
- Poznanie narzędzi i bibliotek, które są obecnie na preferowane przez programistów Ruby on Rails
- Szybkie tworzenie w pełni testowalnych aplikacji internetowych

BOTTEGA IT minds

Szczegółowy program:

1. Wstęp

- 1.1. Środowisko Developerskie
  - 1.1.1. Instalacja - Windows
  - 1.1.2. Instalacja - Linux
  - 1.1.3. RubyMine IDE

2. Aplikacje webowe

- 2.1. Lightweight vs enterprise

3. Zwinne projektowanie

- 3.1. Podejście Agile
- 3.2. User Story
- 3.3. Interakcja z klientem
- 3.4. Automatyzacja testów
  - 3.4.1. Jednostkowe
  - 3.4.2. end2end
- 3.5. Zwinne modelowanie
  - 3.5.1. Elementy Domain Driven Design
    - 3.5.1.1. Bulding Blocks

4. Język Ruby

- 4.1. Filozofia
- 4.2. Różnice pomiędzy dynamicznie a statycznie typowanymi językami programowania
- 4.3. Podstawowe konstrukcje języka (Metody, Klasy, Moduły, Bloki, Sterowanie, Wyrażenia regularne, Yield)
- 4.4. Podstawowe typy (Teksty, Liczby, Tablice, Mapy)
- 4.5. Programowanie zorientowane obiektowo

BO·TT·EGA
IT minds

BO·TT·EGA
IT minds

- 5.3.6.3. Low Coupling
- 5.3.6.4. High Cohesion

5.4. Widoki

- 5.4.1. Rails ActionView
- 5.4.2. Layouty, partiale, helpery
- 5.4.3. Formularze
- 5.4.4. HTML (silnik szablonów: ERB/ HAML - do wyboru)
- 5.4.5. CSS (Sass)
- 5.4.6. Javascript
  - 5.4.6.1. JQuery
- 5.4.7. Coffeescript
  - 5.4.7.1. Ajax
  - 5.4.7.2. Wydajność
- 5.4.8. Best Practices

5.5. Kontrolery

- 5.5.1. Rails ActionController
- 5.5.2. Wzorce Decorator/Presenter i Facade
- 5.5.3. Zasada 'skinny controller, fat model'
- 5.5.4. Obsługa różnych formatów (xml, json, csv, pdf)

5.6. Routing

- 5.6.1. RESTful urls
- 5.6.2. Tworzenie zdalnego API

5.7. Rozsyłanie maili

- 5.7.1. Rails ActionMailer

5.8. Zadania w tle

BOTTEGA IT minds

- 5.8.1. delayed_job
- 5.9. Autentykacja i bezpieczeństwo
  - 5.9.1. Bezpieczne mechanizmy Rails
  - 5.9.2. Zarządzanie użytkownikami przy pomocy devise
  - 5.9.3. Kontrolowanie dostępu
- 5.10. Narzędzia
  - 5.10.1. assets pipeline - kompilator statycznych zasobów
  - 5.10.2. Bundler - zarządzanie zależnościami
  - 5.10.3. Rake - automatyzacja procesu budowania

6. Testowanie

- 6.1. Testowanie jednostkowe
  - 6.1.1. Zalety testowania w izolacji
  - 6.1.2. Podstawowe pojęcia i techniki
  - 6.1.3. Test-Driven Development
    - 6.1.3.1. Narzędzia wspomagające TDD: guard, spork
  - 6.1.4. Testowanie JavaScript
- 6.2. RSpec
  - 6.2.1. Testowanie modeli
  - 6.2.2. Testowanie kontrolerów
  - 6.2.3. Testowanie widoków
  - 6.2.4. Testowanie route'ów
  - 6.2.5. Przygotowanie stanu (fixture, factory)
  - 6.2.6. Mockowanie
- 6.3. Testowanie integracyjne/akceptacyjne
  - 6.3.1. Metodyka Behavior-Driven Development

BO·TT·EGA
IT minds