

Program szkolenia:

## **PHP revisited - odświeżenie spojrzenia na programowanie w PHP**

Informacje:

<b>Nazwa:</b>	<b>PHP revisited - odświeżenie spojrzenia na programowanie w PHP</b>
<b>Kod:</b>	<b>PHP-revisited</b>
<b>Kategoria:</b>	PHP
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	60% wykłady / 40% warsztaty

---

Szkolenie zostało przygotowane z myślą o programistach i architektach PHP, w celu zapoznania ich z aktualnymi technikami stosowanymi w PHP, best-practices czy różnego rodzaju problemami/błędami występującymi przy implementacji aplikacji PHP, jak i sposobami na ich uniknięcia.

## Szczegółowy program:

### 1. Rozwój języka PHP

1.1. Zmiany języka w PHP 4, 5, 7.

1.2. HHVM

1.3. Integracja PHP z serwerem www

1.3.1. Wprowadzenie do tematyki

1.3.2. Przegląd rozwiązań

1.3.2.1. Apache2

1.3.2.2. Lighttpd

1.3.2.3. nginx

1.3.2.4. Built-in webserver PHP 5.4

1.3.3. PHP FastCGI Process Manager

1.3.3.1. Pule procesów

1.3.3.2. Kluczowe aspekty konfiguracji

1.3.3.3. Aspekty wydajnościowe i monitoring usługi

### 2. Standaryzacja PHP Framework Interop Group

2.1. Wprowadzenie do tematyki

2.2. Przegląd zaakceptowanych PSR

2.2.1. PSR-0 Autoloading Standard

2.2.2. PSR-1 Basic Coding Standard

2.2.3. PSR-2 Coding Style Guide

2.2.4. PSR-3 Logger Interface

2.2.5. PSR-4 Autoloading Standard

2.2.6. PSR-7 HTTP Message Interface

## 2.3. Praktyczne zastosowanie PSR w projektach

### 2.3.1. Zarządzanie zależnościami i bibliotekami w projekcie

#### 2.3.1.1. Pakiety PEAR

#### 2.3.1.2. Composer

#### 2.3.1.3. Struktura pliku composer.json

#### 2.3.1.4. Definiowanie zależności projektowych

#### 2.3.1.5. Wykonywanie skryptów przed i/lub po instalacji pakietu

#### 2.3.1.6. Przegląd typowych problemów

#### 2.3.1.7. Aspekty wydajnościowe Composera

#### 2.3.1.8. Udostępnianie pakietów publicznych i prywatnych

#### 2.3.1.9. Packagist

#### 2.3.1.10. Toran Proxy

### 2.3.2. Zarządzanie logami aplikacyjnymi

#### 2.3.2.1. Monolog

#### 2.3.2.2. Koncepcja

#### 2.3.2.3. Handlers

#### 2.3.2.4. Formatters

#### 2.3.2.5. Processors

#### 2.3.2.6. Integracje z frameworkami

### 2.3.3. Zarządzanie dostępem do systemu plików

#### 2.3.3.1. Wpływ wymiany stosowanego storage plików na zmiany w kodzie źródłowym

#### 2.3.3.2. Gaufrette

#### 2.3.3.3. Integracje z frameworkami

## 3. Techniki testowania oprogramowania

### 3.1. Wprowadzenie do tematyki

3.2. Test Driven Development

3.3. Behavior Driven Development

3.4. Przegląd popularnych narzędzi do testowania oprogramowania

3.4.1. PHPUnit

3.4.2. Behat

3.4.3. PHPSpec

3.5. Refaktoryzacja istniejącego kodu

## 4. Debuggowanie i profilowanie aplikacji PHP

4.1. Wprowadzenie do tematyki

4.2. Profilowanie wydajnościowe aplikacji

4.3. Narzędzia

4.3.1. Xdebug

4.3.2. Blackfire PHP

4.3.3. Analiza uzyskanych informacji w aplikacjach kCacheGrind/MCG

4.3.4. Integracja z IDE na przykładzie PHPStorm

## 5. Wzorce projektowe

5.1. Wprowadzenie do tematyki

5.2. Przegląd zastosowań wybranych wzorców

5.2.1. MVC

5.2.2. Dependency Injection

5.2.2.1. Konfiguracja kontenera DI w ramach frameworka full-stack i microframeworka

5.2.3. Factory

5.2.4. Strategy

5.3. Antywzorce projektowe

#### 5.4. Powiązane techniki

5.4.1. SOLID

5.4.2. GRASP

5.4.3. DRY

5.4.4. YAGNI

## 6. Nowoczesne frameworki PHP

### 6.1. Przegląd rozwiązań

#### 6.1.1. Frameworki full-stack

6.1.1.1. Przegląd rozwiązań

6.1.1.2. Symfony 2

6.1.1.3. Laravel

6.1.1.4. Zend Framework

#### 6.1.2. Mikroframeworki PHP

6.1.2.1. FirePHP

6.1.2.2. Silex

6.1.2.3. Slim

#### 6.1.3. Rozwiązania hybrydowe

6.1.3.1. Phalcon, high-performance C++ extension framework

### 6.2. Zastosowanie omówionych w trakcie szkolenia aspektów w nowoczesnych frameworkach

## 7. Przegląd przykładowych problemów występujących w aplikacjach webowych PHP

### 7.1. Ogólna wydajność rozwiązania

7.1.1. Wpływ architektury kodu na szybkość działania aplikacji

7.1.2. Akceleratory kodu

7.1.3. Akceletatory HTTP

### 7.2. Problemy z kodowaniem znaków i internacjonalizacją aplikacji

### 7.3. Skalowalność

#### 7.3.1. Utrzymanie sesji w środowisku rozproszonym

#### 7.3.2. Storage plikowy w środowisku rozproszonym

### 7.4. Assets management

#### 7.4.1. Odświeżanie zasobów CSS/JS/GFX wraz z wydaniem aplikacji

#### 7.4.2. Optymalizacja pobierania zasobów CSS/JS/GFX

#### 7.4.3. Biblioteki

### 7.5. Deployment aplikacji