

## Program szkolenia:

# Tworzenie aplikacji webowych z wykorzystaniem frameworka Django

### Informacje:

<b>Nazwa:</b>	<b>Tworzenie aplikacji webowych z wykorzystaniem frameworka Django</b>
<b>Kod:</b>	<b>python-django</b>
<b>Kategoria:</b>	Python
<b>Grupa docelowa:</b>	developerzy architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% ćwiczenia

Zakres szkolenia obejmuje informacje dotyczące tworzenia aplikacji webowych przy wykorzystaniu frameworka Django. Poza podstawowymi elementami frameworka, omówione zostaną także zagadnienia automatyzacji procesu deploymentu oraz zarządzania infrastrukturą, pomagające w szybkim i powtarzalnym wdrażaniu tworzonych projektów.

Przedstawiony zostanie także stos technologiczny popularnych bibliotek i aplikacji wykorzystywanych przy tworzeniu projektów w frameworku Django. Poruszone zostaną także kwestie bezpieczeństwa, profilowania i wydajności implementowanych rozwiązań.

W trakcie szkolenia zostaną zaprezentowane dobre praktyki tworzenia komponentów aplikacji webowych w frameworku Django z uwzględnieniem automatycznego testowania każdego z powstających elementów projektu.

### Zalety szkolenia:

- Podejście do projektowania architektury aplikacji i systemu
- Pułapki i najlepsze praktyki

## Szczegółowy program:

### 1. Wstęp

- 1.1. Instalacja środowiska
- 1.2. Stworzenie projektu
- 1.3. Struktura projektu
- 1.4. Ustawienia projektu
- 1.5. Skrypt zarządzający manage.py

### 2. Architektura frameworka

### 3. Widoki

- 3.1. Funkcyjne
- 3.2. Klasowe
- 3.3. Widoki generyczne
- 3.4. Definiowanie struktury adresów URL
- 3.5. Dekoratory

### 4. Szablony

- 4.1. Filtry
- 4.2. Tagi
- 4.3. Dziedziczenie
- 4.4. Zawieranie

### 5. ORM

- 5.1. Modele
- 5.2. Zapytania
- 5.3. Managery
- 5.4. Transakcje

5.5. Migracje bazy danych (south)

5.6. Profilowanie

5.7. Optymalizacja

## 6. Panel administratora

## 7. Formy

7.1. Definiowanie form

7.2. Formy modelowe

## 8. Middleware

8.1. Wbudowane middleware

8.2. Tworzenie własnych middleware

## 9. Testowanie

9.1. Jednostkowe

9.2. Integracyjne

## 10. Sygnały

10.1. Odbieranie

10.2. Wysyłanie

## 11. Logowanie

11.1. Konfiguracja

11.2. Obsługa

11.3. Filtrowanie

## 12. Cachowanie

12.1. Konfiguracja

12.2. Cachowanie widoków

12.3. Cachowanie szablonów

12.4. Cachowanie dowolnych danych

### 13. Internacjonalizacja i lokalizacja

### 14. Bezpieczeństwo

14.1. Najpopularniejsze zagrożenia

14.2. Typowe błędy

14.3. Rozwiązania wbudowane w framework

### 15. Popularne biblioteki i aplikacje

15.1. Debugowanie i profilowanie projektu (django-debug-toolbar)

15.2. Migracje bazy danych (south)

15.3. Tworzenie RESTful API (django-rest-framework)

15.4. Logowanie błędów (sentry)

15.5. Wykonywanie zadań w tle (celery)

### 16. Deployment i zarządzanie infrastrukturą

16.1. Produkcyjny stos technologiczny (nginx/uwsgi)

16.2. Automatyzacja procesu deploymentu (fabric)

16.3. Automatyzacja zarządzania konfiguracją (ansible)