Modernizacja aplikacji PHP (kod: PHP-LEGACY)

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Opis i cel szkolenia

Wiele firm utrzymuje systemy w PHP, które powstały kilka albo kilkanaście lat temu. Działają, ale na starej wersji języka, bez Composera, bez testów i z kodem, którego nikt nie chce dotykać. Na tym szkoleniu pokazujemy, jak taki system unowocześniać krok po kroku, bez przepisywania go od zera: zaplanować przejście na PHP 8, uporządkować zależności, wprowadzić analizę statyczną i automatyczną refaktoryzację, a każdą zmianę zabezpieczyć testami.

Po dwóch dniach szkolenia:

- ocenisz stan starszej aplikacji i zaplanujesz jej aktualizację etapami, zamiast jednego „wielkiego skoku”,
- przeniesiesz zależności do Composera i uporządkujesz autoloading,
- wykorzystasz PHPStan do wyłapywania błędów, zanim kod trafi na produkcję,
- przeprowadzisz automatyczne zmiany kodu Rectorem i będziesz wiedzieć, co po nim sprawdzić ręcznie,
- napiszesz testy, które utrwalą obecne działanie systemu i pozwolą go bezpiecznie zmieniać,
- unowocześnisz kod z użyciem możliwości PHP 8: typów, enumów, właściwości readonly i atrybutów.

Pokazujemy też, jak rozsądnie korzystać z asystentów AI przy analizie i przeróbce starszego kodu i dlaczego każdą taką zmianę trzeba potem sprawdzić narzędziami i testami.

Szkolenie jest dla wszystkich, którzy utrzymują lub rozwijają systemy w PHP: programistów, osób przejmujących cudzy kod, liderów zespołów i architektów planujących modernizację, a także administratorów i DevOpsów odpowiedzialnych za aktualizację środowisk PHP. Jeśli chcesz najpierw szybko poznać sam język, zacznij od szkolenia PHP dla programistów innych języków.

Czas trwania

2 dni

Program

1. Starsza aplikacja PHP – od czego zacząć
   - typowe problemy: stara wersja PHP, include zamiast autoloadingu, zmienne globalne, brak testów
   - ocena stanu kodu i zależności, ryzyka aktualizacji
   - strategia modernizacji etapami: kolejność kroków, małe zmiany zamiast przepisywania od zera
2. Środowisko i zależności
   - uruchamianie aplikacji na kilku wersjach PHP (np. w kontenerach)
   - Composer: przeniesienie zależności, wersjonowanie, plik composer.lock
   - autoloading PSR-4 i porządkowanie struktury katalogów
   - sprawdzanie zgodności bibliotek z nowszą wersją PHP
3. Testy jako zabezpieczenie zmian
   - testy charakteryzacyjne: utrwalenie obecnego zachowania systemu
   - pierwsze testy w PHPUnit dla kodu, który nie był pisany z myślą o testach
   - odcinanie zależności od bazy danych, plików i czasu
4. Analiza statyczna – PHPStan

   - konfiguracja, poziomy analizy, plik baseline dla istniejących błędów
   - stopniowe podnoszenie poziomu analizy w projekcie
   - typowe błędy wykrywane przed uruchomieniem kodu
5. Automatyczna refaktoryzacja – Rector
   - zestawy reguł aktualizacji do kolejnych wersji PHP
   - wprowadzanie zmian partiami i przegląd wyników
   - co zrobi narzędzie, a co trzeba poprawić ręcznie
6. PHP 8 z perspektywy migracji
   - zmiany niekompatybilne wstecz i najczęstsze miejsca awarii po aktualizacji
   - deklaracje typów, typy unijne, operator nullsafe, wyrażenie match
   - enumy, właściwości readonly, promocja właściwości w konstruktorze, atrybuty
7. Refaktoryzacja krok po kroku
   - wydzielanie funkcji i klas z długich skryptów
   - wstrzykiwanie zależności zamiast zmiennych globalnych i singletonów
   - standard kodowania i automatyczne formatowanie (PHP CS Fixer lub analogiczne)
8. AI przy modernizacji kodu
   - asystent AI do objaśniania starszego kodu i proponowania zmian
   - weryfikacja zmian: analiza statyczna, testy, przegląd kodu
9. Utrzymanie efektów
   - rzut oka na uruchamianie analizy statycznej i testów w procesie CI
   - plan dalszych kroków po szkoleniu

Dla grup na zamówienie przykłady możemy oprzeć na problemach zbliżonych do systemu zamawiającego, a program rozszerzyć do 3 dni, np. o aktualizację wersji frameworka (Laravel, Symfony), wydzielanie modułów z dużej aplikacji, konteneryzację środowiska (Docker) albo modernizację warstwy dostępu do bazy danych.

Przeznaczenie i wymagania

Szkolenie jest przeznaczone przede wszystkim dla:
- programistów PHP utrzymujących i rozwijających starsze aplikacje,
- osób przejmujących istniejący system w PHP,
- liderów zespołów i architektów planujących modernizację,
- administratorów i DevOpsów aktualizujących środowiska PHP.

Od uczestników oczekujemy:
- praktycznej znajomości PHP, w tym podstaw programowania obiektowego,
- ogólnej orientacji w pracy z wierszem poleceń i Gitem.

Znajomość Composera, PHPStan czy PHPUnit nie jest wymagana – tego uczymy na szkoleniu.

Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.

Lokalizacje

- Warszawa – ul. Jasna 14/16A
- Zdalnie – zajęcia realizowane poprzez platformę Zoom
- Kraków – ul. św. Filipa 23
- Warsaw (English) – Jasna 14/16A
- Online (English) – your home, office or wherever you want

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

— na życzenie dowolne miejsce w Polsce, lub UE (zajęcia prowadzone w języku angielskim)

Cena szkolenia

2290 PLN netto (VAT 23%)

W cenę szkoleń organizowanych w naszej siedzibie wliczone są:

— autorskie materiały szkoleniowe,
— indywidualne stanowisko komputerowe do pracy podczas zajęć,
— certyfikaty ukończenia szkolenia,
— drobny poczęstunek oraz ciepłe i zimne napoje,
— możliwość jednorazowego kontaktu z instruktorem (instruktorami) po szkoleniu i zadawania pytań dotyczących materiału szkolenia.

Cena szkolenia nie zawiera obiadów. Można je dokupić w cenie 35 zł netto za obiad.

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl