PROGRAM SZKOLENIA DOBRE PRAKTYKI

Szkolenie Architektura frontendu

Zaprojektujesz frontend dla kilku zespołów i wyjdziesz z planem zmian dla swojej aplikacji.

CZAS TRWANIA	POZIOM	FORMA
3 dni	**zaawansowany**	**warsztat dla zespołu, zdalnie lub stacjonarnie**

Mały projekt da się utrzymać w każdej strukturze. Problem zaczyna się, gdy nad jednym frontendem pracuje kilka zespołów: zmiana w jednym miejscu psuje drugie, build trwa coraz dłużej, a nikt nie wie, kto odpowiada za wspólny komponent.

Na szkoleniu rozłożysz architekturę dużej aplikacji frontendowej na decyzje, które da się uzasadnić: gdzie renderować, jak dzielić kod, kiedy monorepo, kiedy mikrofrontendy, a kiedy żadne z nich. Pracujesz na przykładach z prawdziwych projektów. Przed szkoleniem wysyłasz krótki opis architektury swojego projektu (bez kodu objętego NDA), a trzeci dzień kończysz planem zmian dla swojej aplikacji.

PROGRAM

3 dni

- Po co architektura?
  - Objawy: długi build, konflikty, efekt domina po zmianach
  - Architektura a struktura zespołów (prawo Conwaya)
  - Decyzje zapisane w ADR (Architecture Decision Records)
- Gdzie renderować
  - CSR, SSR, SSG i streaming: koszty i zyski każdego podejścia
  - React Server Components (i odpowiedniki w innych frameworkach) oraz granica między serwerem a klientem
  - Architektura wysp (Astro) i częściowa hydratacja
  - BFF (Backend for Frontend) jako warstwa dla interfejsu
  - Ćwiczenie: mapa renderowania dla przykładowej aplikacji
- Podział kodu
  - Podział według funkcji zamiast typów plików
  - Warstwy: interfejs, logika, dostęp do danych
  - Publiczne API modułu i reguły importów
  - Pilnowanie granic w linterze: `@nx/enforce-module-boundaries`, eslint-plugin-boundaries
  - Graf zależności i cykle w dependency-cruiser
  - Ćwiczenie: reguły granic i wykrycie cyklu w przykładowym repo

- Monorepo
  - Kiedy monorepo, a kiedy osobne repozytoria
  - Nx i Turborepo: zadania, cache, zależności między paczkami
  - Workspaces w pnpm i npm
  - Wspólne biblioteki: komponenty, konfiguracja, typy
  - Ćwiczenie: ocena konfiguracji monorepo na gotowym przykładzie
- Mikrofrontendy
  - Kiedy mają sens, a kiedy tylko komplikują
  - Integracja w czasie budowania i w czasie działania
  - Module Federation 2.0 przez `@module-federation/enhanced` (webpack i Rspack; Rspack ma wbudowaną wersję 1.5), w Vite przez `@module-federation/vite`
  - Alternatywy: import maps i Web Components
  - Wspólny routing, stan i design system
- Stan i dane
  - Stan serwera a stan interfejsu
  - Gdzie trzymać stan w dużej aplikacji
  - Kontrakty API i typy generowane ze schematu jako decyzja zespołów
- Design system
  - Biblioteka komponentów i jej wersjonowanie
  - Tokeny projektowe w standardzie DTCG (pierwsza stabilna wersja 2025.10)
  - Kto utrzymuje design system: zespół platformowy czy wspólnota
- Jakość i ewolucja architektury
  - Podział odpowiedzialności za testy między zespoły
  - Budżety wydajności jako kryterium architektury
  - Refaktoryzacja etapami: wzorzec strangler fig i feature flagi
  - Warsztat: plan zmian dla Twojego projektu

Szczegóły techniczne testów, wydajności i pracy ze starym kodem znajdziesz na szkoleniach Strategia testowania, Wydajność aplikacji i Legacy Code.

DLA KOGO?

- Dla tech leadów i architektów odpowiedzialnych za frontend.
- Dla zespołów, w których nad jedną aplikacją pracuje kilka grup.
- Dla programistów, którzy planują przebudowę dużego projektu.

CO MUSISZ UMIEĆ?

- Pracować z nowoczesnym frameworkiem frontendowym w dużym projekcie.
- Pisać w TypeScript i konfigurować narzędzia do budowania aplikacji.

CZEGO SIĘ NAUCZYSZ?

- Wybierzesz sposób renderowania (CSR, SSR, SSG, Server Components, wyspy) dla każdej części aplikacji.
- Podzielisz aplikację na moduły z jasnymi granicami i publicznym API.
- Ustawisz automatyczne pilnowanie granic modułów w linterze.
- Ocenisz, czy Twój projekt potrzebuje monorepo lub mikrofrontendów.
- Przejrzysz konfigurację monorepo z cache zadań i wskażesz, co poprawić.
- Zdecydujesz, gdzie trzymać stan serwera i stan interfejsu w aplikacji kilku zespołów.
- Zaplanujesz design system oparty na tokenach projektowych.
- Zapiszesz decyzje architektoniczne tak, żeby zespół je rozumiał.
- Przygotujesz plan przebudowy istniejącej aplikacji małymi krokami.

OPINIE UCZESTNIKÓW

> *„Piotr ma bardzo wysokie kompetencje w temacie Frontend ogólnie, a szczególnie w frameworku Angular."*
>
> Artur, Software Engineer (kilka szkoleń)

> *„Before I got to know Piotr, I had considered myself pretty apt and up-to-date with latest developments in that realm, but it didn't take long for me to realise that my fellow team member was on a whole different level!"*
>
> Aleksander Przeszło, współpracownik z zespołu w Box (rekomendacja na LinkedIn)

TRENER

Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

KONTAKT piecioshka.pl/kontakt
E-MAIL piecioshka+blog [małpka] gmail.com
SZKOLENIE piecioshka.pl/szkolenia/architektura-frontendu/