PROGRAM SZKOLENIA APLIKACJE WEBOWE

Szkolenie API: REST i GraphQL

Zaprojektujesz API w REST i GraphQL, z którego inne zespoły skorzystają bez dopytywania.

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

Dobre API to umowa między zespołami. Jeśli jest czytelne, przewidywalne i dobrze opisane, integracja z nim zajmuje godziny, a nie tygodnie. Na szkoleniu projektujesz i budujesz API w Node.js: najpierw w stylu REST, potem w GraphQL, a na koniec łączysz je z innymi systemami. To szkolenie o projektowaniu kontraktu, a nie o frameworku: te same zasady przeniesiesz do Javy czy .NET.

Dużo miejsca zajmuje to, co wychodzi dopiero na produkcji: błędy, ponawianie zapytań, limity i zmiany wersji, które nie psują klientów.

PROGRAM

3 dni

- Protokół HTTP w praktyce
  - Metody i ich semantyka: bezpieczne i idempotentne
  - Nagłówki: `Content-Type`, `Accept`, `Cache-Control`, `ETag`
  - Kody odpowiedzi: `200`, `201`, `204`, `400`, `401` a `403`, `404`, `409`, `422`, `429`
- Projektowanie zasobów REST
  - Nazwy zasobów, relacje i zagnieżdżanie adresów
  - Filtrowanie, sortowanie i paginacja (offset i kursor)
  - Wersjonowanie API i wycofywanie: nagłówki `Deprecation` i `Sunset`
- Modelowanie i walidacja danych
  - JSON: typy, daty, liczby i wartości `null`
  - JSON Schema jako źródło prawdy o kształcie danych
  - Walidacja wejścia i wyjścia na serwerze
- Obsługa błędów
  - Jednolity format błędów według RFC 9457 (Problem Details)
  - Błędy walidacji przypisane do konkretnych pól
- Dokumentacja i kontrakt
  - OpenAPI 3.1 i 3.2: opis endpointów, schematów, przykładów i strumieni
  - Podejście contract-first i generowanie klientów
  - Interaktywna dokumentacja dla innych zespołów

- GraphQL
  - Schemat, typy, zapytania i mutacje
  - Resolvery i kontekst zapytania
  - Problem N+1 i DataLoader
  - Ewolucja schematu: `@deprecated` zamiast wersjonowania
  - Ograniczanie głębokości i kosztu zapytań (pokaz)
- Uwierzytelnianie i autoryzacja
  - Klucze API, tokeny Bearer i JWT
  - OAuth 2.0: Authorization Code z PKCE według RFC 9700 (Security BCP)
  - OpenID Connect: logowanie użytkownika a dostęp do API
  - Uprawnienia na poziomie zasobu i pola: BOLA i BOPLA z OWASP API Security Top 10
- Integracje z innymi systemami
  - Webhooki: podpisy, ponowienia, kolejność zdarzeń i deduplikacja po stronie odbiorcy
  - WebSockets i Server-Sent Events do powiadomień na żywo
  - Ponawianie zapytań z backoffem i klucze idempotencji
- Ochrona API
  - Rate limiting: `429`, `Retry-After` i nagłówki `RateLimit` oraz `RateLimit-Policy` (szkic IETF)
  - CORS i limity rozmiaru zapytań
- Testowanie API
  - Testy integracyjne endpointów
  - Testy kontraktowe na podstawie OpenAPI
  - Ręczne sprawdzanie zapytań z linii poleceń i w klientach HTTP

Bezpieczeństwo szerzej omawia szkolenie Bezpieczeństwo WWW, a komunikację na żywo szkolenie WebSockets.

DLA KOGO?

- Dla programistów, którzy budują lub rozwijają API.
- Dla programistów frontendu, którzy chcą rozumieć, jak powstaje API.
- Dla zespołów, które integrują kilka systemów i tracą dni na niejasne błędy i zerwane kontrakty.

CO MUSISZ UMIEĆ?

- Programować w JavaScript i znać podstawy Node.js.
- Rozumieć, czym jest zapytanie HTTP i format JSON.

CZEGO SIĘ NAUCZYSZ?

- Dobierzesz metodę i kod odpowiedzi HTTP tak, żeby klient wiedział, co się stało.
- Zaprojektujesz zasoby REST z paginacją, wersjonowaniem i czytelnymi błędami, a stare pola wycofasz bez psucia klientów, także w GraphQL.
- Opiszesz API w OpenAPI i zwalidujesz dane przez JSON Schema.

- Zbudujesz serwer GraphQL bez problemu N+1.
- Zabezpieczysz API tokenami, OAuth 2.0 z PKCE i uprawnieniami do zasobów, a logowanie oprzesz na OpenID Connect.
- Połączysz systemy webhookami z podpisami, ponowieniami i odpornością na duplikaty.
- Ochronisz API limitem zapytań i przetestujesz jego kontrakt.

OPINIE UCZESTNIKÓW

> *„Gruntownie, krok po kroku stworzyliśmy aplikację sklepu internetowego, po drodze pisząc testy jednostkowe, e2e, web komponenty, autorski backend oparty na Node.js + Express.js, jak również wykorzystaliśmy narzędzia CI."*
>
> Łukasz, JavaScript Developer (WarsawJS MasterClass)

> *„Piotr przekazywał wiedzę na temat tworzenia odpornych na błędy aplikacji internetowych. Podczas szkolenia tworzyliśmy testy jednostkowe, testy end-to-end aplikacji po stronie klienta oraz testy backendu napisanego w Node.js."*
>
> Dominik, Frontend Developer (WarsawJS MasterClass)

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/api/**