



SYLABUS PRZEDMIOTU

Nazwa przedmiotu: Trendy w rozwoju API

Kod przedmiotu: TAPI

Kierunek / Profil: Informatyka / praktyczny

Tryb studiów: stacjonarny

Rok / Semestr: 4 / 7

Charakter: obowiązkowy

Odpowiedzialny: mgr Mateusz Miotk

Wersja z dnia: 19.02.2026

1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratoria	Z prowadzącym	Praca własna	Łącznie	ECTS
30 h	30 h	—	60 h	65 h	125 h	5

2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Laboratorium	Zaliczenie z oceną
Wykład	Egzamin

3. Cel dydaktyczny

Przedmiot "Trendy w rozwoju API" ma na celu zaznajomienie studentów z aktualnymi kierunkami rozwoju i innowacjami w dziedzinie tworzenia API dla aplikacji internetowych. Studenci zdobędą wiedzę teoretyczną i praktyczną na temat najnowszych standardów, protokołów i narzędzi używanych do projektowania, implementacji i zarządzania interfejsami API. Uczestnicy będą mieli okazję zgłębić tematykę takich kwestii jak: bezpieczeństwo

API, skalowalność, wydajność oraz dobre praktyki programistyczne. Poznają również różnorodność stylów API, w tym REST, GraphQL, gRPC i inne, analizując ich zalety i wady oraz zastosowania w różnych kontekstach projektowych. Kolejnym istotnym elementem przedmiotu jest praca grupowa, która ma na celu rozwijanie umiejętności współpracy, komunikacji i innowacyjnego myślenia. Studenci będą mieli za zadanie zanalizować i opracować materiał dotyczący wybranej technologii z zakresu tworzenia API, co pozwoli na pogłębienie wiedzy i umiejętności praktycznych w konkretnym obszarze.

4. Przedmioty wprowadzające

Przedmiot	Wymagane zagadnienia
Technologie Frontendowe	Technologie Backendowe
Technologie internetu	Umiejętność posługiwania się emulatorem terminala w systemie operacyjnym
Znajomość struktur plików i katalogów w systemie operacyjnym	Znajomość pojęcia procesu w systemie operacyjnym
Znajomość protokołu HTTP	Znajomość pojęć z zakresu tworzenia aplikacji internetowej z podziałem na frontend oraz backend.

5. Treści programowe

1. Wprowadzenie do API (Application Programming Interface). Przegląd różnych typów API (REST, SOAP, GraphQL, gRPC, itd.). Znaczenie i role API w rozwoju aplikacji internetowych
2. Konwencje, normy i najlepsze praktyki w tworzeniu REST API
3. GraphQL - wprowadzenie i porównanie z REST. Tworzenie prostych API z wykorzystaniem GraphQL
4. Wprowadzenie do gRPC. Porównanie gRPC z REST i GraphQL, zalety i wady
5. Rola i znaczenie dokumentacji API. Narzędzia do tworzenia dokumentacji API - OpenAPI, Swagger, Postman
6. Język TypeScript
7. Inne znane technologie wspierające proces budowania aplikacji internetowych.

6. Efekty kształcenia

Wiedza

- Student rozumie podstawowe koncepcje i zasady rządzace współczesnymi API, w tym ich role i zastosowania w różnych typach aplikacji internetowych.
- Student rozumie metody zapewniania jakości i testowania API, włączając w to testy jednostkowe, integracyjne oraz testy obciążenia.
- Student rozumie kwestie związane z zabezpieczaniem API, w tym uwierzytelnianie, autoryzację oraz ochronę przed powszechnymi zagrożeniami i atakami.
- Student rozumie etyczne i prawne aspekty tworzenia i zarządzania API, w tym prywatność, ochronę danych i zgodność z regulacjami prawnymi.

Umiejętności

- Student umie projektować interfejsy API, z wykorzystaniem odpowiednich metod, protokołów i standardów, aby zapewnić ich wydajność, bezpieczeństwo i skalowalność.
- Student umie analizować i oceniać różne style i architektury API (takie jak REST, GraphQL, gRPC), rozumiejąc ich zalety, wady i najlepsze zastosowania.
- Student umie efektywnie komunikować się w zespole programistycznym, współtworzyć dokumentację API i udostępniać informacje o API zespołowi i interesariuszom.
- Student umie wykorzystywać narzędzia i technologie wspierające proces tworzenia, wdrażania i monitorowania API, zwiększać efektywność i jakość pracy.
- Student umie projektować zaawansowane interfejsy API, z wykorzystaniem odpowiednich metod, protokołów i standardów, aby zapewnić ich wydajność, bezpieczeństwo i skalowalność.

Kompetencje społeczne

- Student jest gotów do jest gotów do samodzielnego uczenia się przez całe życie
- Student potrafi pracować w zespole nad wspólnym projektem związanym z tworzeniem API, wykazując się umiejętnościami analitycznymi, kreatywnymi i komunikacyjnymi.

7. Kryteria oceny

- wykład z elementami dyskusji z prezentacją multimedialną
- burza mózgów
- rozwiązywanie zadań
- analiza przypadków
- projekt praktyczny
- Kryteria oceny
- 50% Projekt programistyczny
- 50% Prezentacja projektu oraz dokumentacji
- 40% Ocena z laboratorium
- 60% Egzamin z zagadnień poruszanych na wykładzie

8. Metody dydaktyczne

Wykład, laboratoria, praca własna studenta.

9. Literatura

Podstawowa:

- M. Amundsen, RESTful Web API Patterns and Practices Cookbook, O'Reilly Media, 2023.
- S. Buna, GraphQL in Action, Manning Publications, 2021.
- A. Freeman, Essential TypeScript, Apress, 2019.
- J. S. Ponelat, Designing APIs with Swagger and OpenAPI, Manning Publications, 2021.

Uzupełniająca:

- F. Dogilo, REST API Development with Node.js, Apress, 2018.
- JJ. Geewax, API Design Patterns, Manning Publications, 2021.