

## Zadanie - Pierwsze API

Zadanie polega na zbudowaniu prostego API w Pythonie, połączenia go z bazą MySQL (XAMPP) oraz odebraniu danych na stronie internetowej.

### Krok 1. Utworzenie projektu z FastAPI (na 2)

- Utwórz projekt w programie PyCharm
- Zainstaluj biblioteki: fastapi, uvicorn, mysql-connector-python
- Wykorzystaj kod startowy main.py do utworzenia API oraz endpointa /start
- Uruchom projekt i w przeglądarce wejdź na adres <http://127.0.0.1:8000/start>
- Zmień tekst zwracany przez endpoint /start na własny (np. swoje imię i nazwisko)

### Krok 2. Baza danych w MySQL (XAMPP) i połączenie z Pythonem (na 3)

- Uruchom XAMPP, przejdź do pypMyAdmin i utwórz nową bazę danych
- Utwórz tabelę np. Items (przedmioty, gry, filmy, ubrania, itp. - tematyka dowolna)
- Dodaj kilka przykładowych rekordów
- Wykorzystaj przykładowy kod z pliku db\_test.py, aby połączyć się z bazą danych i wykonać SELECT
- Wypisz w konsoli Pythona rekordy pobrane z tabeli za pomocą SELECTa

### Krok 3. Endpoint /items - pobieranie danych z API (na 4)

- Do pliku main.py dodaj nowy endpoint (@app.get("/items")) [wykorzystaj kod endpointa /start]
- Nowy endpoint powinien:
  - Połączyć się z bazą
  - Wykonać SELECT z tabeli (kod z kroku 2)
  - Zwrócić dane w postaci JSON
- Sprawdź działanie endpointa w przeglądarce (<http://127.0.0.1:8000/items>)

### Krok 4. Frontend (na 5)

- Utwórz prostą stronę w HTML i CSS
- Dodaj w <script> kod JS, który po naciśnięciu przycisku
  - wykona fetch("http://127.0.0.1:8000/items") i odbierze JSON
  - wstawi odebrane dane do tabeli
- Wykorzystaj przykładowy kod funkcji fetch() z pliku fetch.js*
- Otwórz stronę w przeglądarce i sprawdź czy tabela się wypełnia

### Krok 5. Wysyłanie danych z formularza za pomocą POST (na 6)

- Dodaj do strony formularz z polami jak w tabeli w bazie danych
- W JS zbierz dane z pól i wyślij je do API z metodą POST (przykład: post.js)
- W main.py dodaj endpoint, który:
  - odbiera dane (przykład: endpoint\_post.py)
  - Wstawia rekord do bazy danych (INSERT)
  - Zwraca potwierdzenie (np. {"status": "ok", "id": "...", ...})
- Na froncie wyświetl alert (np. „Dodano przedmiot”) i odśwież tabelę (== wywołaj ponownie funkcję z przycisku)

### Na dodatkowe oceny:

- Dodaj endpoint obliczający coś na podstawie przesłanych parametrów (np. liczba sztuk i id przedmiotu) i danych z bazy (np. cena)
- Zmodyfikuj endpoint swojego API, aby wykonał połączenie z endpointem zewnętrznego API, pobrał dane i wykorzystał je w działaniu  
*np. NBP Web API - oficjalne API Narodowego Banku Polskiego, przykład dla kursu EUR-PLN:*  
<https://api.nbp.pl/api/exchangerates/rates/a/eur/>