

13.05.2022

Na dzisiejszych zajęciach zapoznamy się z architekturą serwisów sieciowych REST (*Representational State Transfer*) na przykładzie serwisu [Discogs](#).

- Zapoznaj się z witryną [Discogs](#) oraz ze [specyfikacją udostępnianego przez nią REST API](#).
- Spróbuj wykonać kilka przykładowych zapytań o dane z bazy Discogs przy pomocy narzędzia [curl](#). Zwróć uwagę, że dokumentacja REST API Discogs określa, jak należy ustawić nagłówek HTTP [User-Agent](#).
- Spróbuj napisać program wyszukujący w bazie Discogs wszystkie albumy danego wykonawcy (np. Budki Suflera). Sprawdź czy to, co drukuje program zgadza się z tym, co można ręcznie wyszukać na witrynie Discogs.

Uwaga 1.: Może się okazać, że Discogs odrzuca zapytania jeśli są one całkowicie anonimowe. Sprawdź wtedy w dokumentacji kim/czym trzeba być, aby uzyskać odpowiedź — zalogowanym użytkownikiem witryny, aplikacją napisaną przez zarejestrowanego dewelopera?

Uwaga 2.: Dla bazy Discogs, podobnie jak dla wielu innych serwisów z REST API, dostępne są biblioteki dla różnych języków programowania "opakowujące" komunikację z serwisem. Jednak **celem tych i następnych ćwiczeń jest użycie REST API bezpośrednio, czyli poprzez wykonywanie odpowiednich zapytań HTTP i bez użycia pomocniczych bibliotek** specyficznych dla danego serwisu.

- Poszukaj innych dyskograficznych baz danych z publicznym API. Porównaj je z Discogs — czy ich API też są zgodne z zasadami REST? Jakie rodzaje obiektów są przetwarzane? Jak wyglądają URL-e odpowiadające tym obiektom, jakie operacje można na nich wykonywać? Kto ma prawo wywoływać poszczególne zdalne operacje?
- Czy powszechnie znane witryny takie jak Gmail, Facebook, Twitter, itd. udostępniają jakieś webAPI pozwalające aplikacjom wykonywać te same operacje, które potrafi wykonywać użytkownik korzystający z przeglądarki? Czy te webowe API są zaprojektowane zgodnie z zasadami architektury REST (czy są to tzw. RESTful APIs)?