Rest API

PHP i MySQL

2024/2025

Zadanie polega na przygotowaniu skryptu w języku **PHP**, który będzie realizował założenia **Rest API**, oraz aplikacji klienta która będzie korzystała z przygotowanego **Rest API**.

Serwer

Magazynem danych po stornie serwera będzie baza danych **MySQL**.

Połączenie z **Rest API** powinno posiadać system zabezpieczenia przed dostępem dla niezalogowanego użytkownika. Tylko zalogowani użytkownicy będą otrzymywać dane z **Rest API**.

Serwer powinien pomimo posiadania autorskiego skryptu API posiadać także utworzoną bazę danych oraz wykorzystywać stworzone API do wymiany danych pomiędzy klientem, a tą bazą danych.

Rest API powinno obsługiwać podstawowe zapytania **CRUD** takie jak:

- **GET**
- **POST**
- **PUT**
- **DELETE**

Rest API powinno zwracać dane w formacie **JSON** oraz odpowiedni **kod odpowiedzi HTTP**.

Adres do **Rest API** powinien składać się z punktu wejścia wraz z określeniem wersji API

np. *mojastrona.pl/api/v1*

Rest API powinno pozwalać na zastosowanie identyfikatorów w ścieżce

np. *mojastrona.pl/api/v1/user/:id*

np. *mojastrona.pl/api/v1/user/123*

gdzie wartość *:id* jest identyfikatorem którego wartość jest dynamiczna.

Kolejną możliwością jest dodawanie opcjonalnych parametrów przy pomocy **query string**

np. *mojastrona.pl/api/v1/user/:id?:nazwa=:wartosc1,:wartosc2*

np. *mojastrona.pl/api/v1/user/123?pola=imie*

np. *mojastrona.pl/api/v1/post/542?pola=tytul,data*

gdzie *:nazwa* określa nazwę parametru, a następnie wartość lub wartości oddzielone przecinkiem, kolejne parametry oddzielane są znakami apmersand **&**

np. *mojastrona.pl/api/v1/post/542?pola=tytul,data&wiersze=1,3,4,5,10*

Ostatnią składową adresu punktu końcowego będzie **operator** czyli dodatkowa wartość która mówi o typie porównania wartości dla parametru. Zapis takiego operatora będzie wymagał zastosowania odpowiedniego sposoby ponieważ składnia **query string** nie pozwala na przekazanie trzeciego elementu. Stosowane są zapisy operatorów w nawiasach kwadratowych lub po dwukropku po nazwie parametru

np. *mojastrona.pl/film?dlugosc[wieksza]=120&ocena[mniejsza]=3.0*

np. *mojastrona.pl/film?dlugosc:wieksza=120&ocena:mniejsza=3.0*

Oczywiście nazwy operatorów są także ustandaryzowane ale nie jest konieczne stosowanie tej konwencji. Sposób zapisu operatorów jest dowolny i to twórca skryptu Rest API określa jak w jego przypadku ma wyglądać poprawnie dodany operator.

Poniżej przedstawiono kilka ustandaryzowanych nazwy dla operatorów:

Operator	Opis
:gt	Większe
:gte	Większe i równe
:lt	Mniejsze
:lte	Mniejsze i równa
:eq	Równe (operator domyślny)
:not	Różne

Brak podania operatora zawsze będzie obsługiwane jako operator równy.

np. *mojastrona.pl/film?dlugosc:gt=120&ocena:lt=3.0*

Oczywiście unikamy stosowania polskich znaków w adresach.

Dane wysyłane do serwera z klienta ze względów bezpieczeństwa i ograniczeń długości adresu powinny być wysyłane za pomocą **request body** w formacie JSON.

Klient

Przygotuj przykładową aplikację klienta w dowolnej technologii, która będzie wykonywała połączenia do napisanego Rest API. Aplikacja powinna przedstawiać metodę logowania się do Rest API, oraz pobieranie, dodawanie, aktualizowanie oraz usuwanie danych przy pomocy Rest API. Sama aplikacja powinna posiadać prosty i przejrzysty interfejs graficzny.

GIT i Docker

Dodatkowo punktowane będzie:

- umieszczenie aplikacji serwera i klienta na GitHubie wraz z opisem działania
- użycie konteneryzacji Dockera do szybkiego uruchomienia developerskich serwerów PHP, MySQL oraz aplikacji klienta.

Pomoce

Link do prezentacji na temat Rest API:

https://www.youtube.com/watch?v=P9b8-BrWdYs&list=PLjHmWifVUNMLjh1nP3p-U0VYrk_9aXVjE