

Warsztaty III

REST

v 1.1

Cel warsztatów

Celem warsztatów jest napisanie pełnej i funkcjonalnej aplikacji do wypożyczania książek metodą REST.

Projekt składa się z dwóch części:

- **Serwer** – napisany w PHP, implementujący funkcjonalność REST,
- **Klient** – napisany w HTML-u i JavaScriptcie, komunikujący się z serwerem za pomocą AJAX.

Serwer ma implementować klasę Book mającą swój identyfikator, nazwę, autora i opis.

Cel warsztatów

Serwer ma też implementować ścieżki dostępu podane w tabeli.

Dla uproszczenia będziemy używać normalnych ścieżek a nie ich wersji /book/1.

O tworzeniu ładniejszych ścieżek dla swoich aplikacji dowiedzie się na następnych warsztatach.

Metoda HTTP	ADRES	CO ROBI?
GET	api/books.php	Zwraca listę wszystkich książek.
POST	api/books.php	Tworzy nową książkę na podstawie danych przekazanych z formularza i zapisuje ją do bazy danych.
GET	api/books.php?id=1	Wyświetla informacje o książce o podanym id.
PUT	api/books.php?id=1&Name=""&autor=""	Zmienia informacje o książce o podanym id na nową.
DELETE	api/books.php?id=1	Usuwa książkę o podanym id z bazy danych.

Cel warsztatów

- Klient ma implementować tylko stronę główną.
- Strona ta ma pokazać wszystkie książki stworzone w systemie. Dane mają być wczytane AJAX-em z **api/books.php**.
- Na górze tej strony ma być też formularz do tworzenia nowych książek wysyłający dane AJAX-em (metoda POST).
- Gdy użytkownik kliknie na nazwę książki, pod nią ma się rozwijać **div** z informacjami na temat tej strony wczytane za pomocą AJAX (GET) z endpointu **api/books.php?id=....**
Div ten ma też zawierać formularz służący do edycji tej książki (AJAX, metoda PUT na endpointzie **api/books.php?id=1&Name=""&autor=""**).
- Obok nazwy ma się znajdować guzik służący do usuwania książki (AJAX, metoda DELETE na endpoint **api/books.php?id=1**)

Pod koniec zajęć wyślij głównemu wykładowcy informację z adresem repozytorium, na którym znajduje się twój kod, listą zaimplementowanych funkcjonalności. Dołącz również zrzut bazy danych.

Zadanie rozgrzewkowe

➤ Za pomocą AJAX-u pobierz aktualną datę z endpointu <http://date.jsontest.com>.

➤ Stwórz w PHP tablicę kolorów i zamień ją na JSON (wyświetl ją na stronie). Napisz stronę HTML, na której za pomocą jQuery wczytasz tablicę kolorów ze swojej strony PHP i wyświetlisz ją na stronie.

Zadanie 1

Przygotowanie

- Przygotuj folder pod aplikację.
 - Załóż nowe repozytorium Git na GitHubie i nową bazę danych.
 - Pamiętaj o robieniu backupów bazy danych (najlepiej co każde ćwiczenie) i tworzeniu **commitów** (też co ćwiczenie).
- Stwórz plik .gitignore i dodaj do niego wszystkie podstawowe treści (pliki *.*~, katalog .idea etc).
 - Stwórz plik, który będzie służył do łączenia się z bazą danych (w katalogu <katalog główny projektu>/api/src).

Zadanie 2

Stworzenie klasy Book

- W katalogu stwórz plik book.php w katalogu: **<katalog główny projektu>/api/src**
- W tym katalogu stwórz klasę Book.
- Stwórz do niej tabelkę w bazie danych.
- Ręcznie wprowadź kilka książek do bazy danych.

Ma ona implementować następujące funkcjonalności:

- SET i GET do wszystkich atrybutów (do id tylko GET).
- Konstruktor nastawiający id na -1, a resztę danych zerujący.
- Funkcję loadFromDB(conn, id).
- Funkcję create(conn, name, autor).
- Funkcję update(conn, name, autor).
- Funkcję delteFromDB(conn).

Ręcznie wprowadź kilka książek do bazy danych.

Zadanie 3

Stworzenie punktu dostępu do wypisania wszystkich książek

- W katalogu **<katalog główny projektu>/api** stwórz plik books.php.
- Jeżeli dane są do niej przekazywane za pomocą GET, to chcemy wypisać wszystkie książki.
- Wczytaj id wszystkich książek, stwórz ich obiekty, a potem zserializuj je do formatu JSON i wyświetl na stronie.

Zadanie 4

Stworzenie strony głównej

- W katalogu **<katalog główny projektu>/** stwórz plik index.php, a w nim podstawowy layout swojej strony.
- Użyj JQuery do stworzenia zapytania AJAX do endpointu api/books.php metodą GET.
- Po otrzymaniu wszystkich danych w formacie JSON wyświetl wszystkie tytuły na stronie.
- Pod każdym tytułem zostaw pusty **div**, w którym później będziemy w nim wyświetlać informacje na temat książki.

Zadanie 5

Stworzenie punktu dostępu do wypisania danych o jednej książce

Przejdź do pliku books.php w katalogu <katalog główny projektu>/api.

- Jeżeli dane są do strony przekazywane za pomocą GET, sprawdź, czy przekazany jest **parametr &_GET['id']**. Jeżeli tak jest, to chcemy przekazać informację tylko o jednej książce (tej o podanym **id**).
- Stwórz obiekt danej książki i wyświetl na stronie jej zserializowany obiekt (na razie tylko tytuł).

Zadanie 6

Strona główna

W katalogu <katalog główny projektu>/ w pliku index.php:

- Do każdego tytułu książki podepnij funkcję click.
- Funkcja ta ma rozwinąć pusty **div** znajdujący się pod książką, wysłać zapytanie AJAX do serwera na **endpoint** `api/books.php?id=1` i wyświetlić odebrane informacje.

Zadanie 7

Stworzenie punktu dostępu do tworzenia nowej książki

W katalogu **<katalog główny projektu>/api** w pliku books.php:

- Jeżeli dane są do strony przekazywane za pomocą POST, to chcemy stworzyć nową książkę.
- Stwórz obiekt nowej książki korzystając z danych z `$_POST`.
- Wyślij informację o sukcesie.

Zadanie 8

Strona główna

W katalogu **<katalog główny projektu>/** w pliku index.php:

- Stwórz na górze strony formularz służący do tworzenia nowej książki.
- Ma on mieć podpiętego AJAX-a, który metodą POST wyśle dane na **endpoint api/books.php**.

Po otrzymaniu wiadomości o udanym dodaniu ma on przeładować od nowa wszystkie książki (żeby pokazać tę nowo dodaną).

Zadanie 9

Stworzenie punktu dostępu do usuwania książki

W katalogu **<katalog główny projektu>/api** w pliku books.php:

- Jeżeli dane są do niej przekazywane za pomocą DELETE, to chcemy usunąć książkę.
- Wczytaj obiekt książki o podanym id, a potem usuń go z bazy danych.

Zadanie 10

Strona główna

W katalogu **<katalog główny projektu>/** w pliku index.php:

- Obok każdej książki wyświetl link służący do usunięcia książki.
- Podepnij do niego funkcję **click**.
- Funkcja ta ma przesyłać AJAX-em zapytanie DELETE na **endpoint api/books.php?id=...**
- Potem ma załadować od nowa wszystkie książki (żeby nie wyświetlać usuniętej).