

Dokumentacja API JSONPlaceholder

Autor: Mateusz Siudek

Grupa: ASI_L5

Informatyka rok 3

Opis API

API JSONPlaceholder to uproszczony serwis do zarządzania postami, działający lokalnie.

Dane początkowe są pobierane z publicznego API **JSONPlaceholder**

(<https://jsonplaceholder.typicode.com/posts>) podczas startu aplikacji i przechowywane w pamięci aplikacji. API umożliwia:

- Pobieranie listy wszystkich postów
- Pobieranie szczegółów pojedynczego posta na podstawie ID
- Dodawanie nowych postów
- Aktualizowanie istniejących postów
- Usuwanie postów

Uwaga: Dane są przechowywane wyłącznie w pamięci aplikacji. Po ponownym uruchomieniu serwera wszystkie zmiany zostaną utracone, a dane zostaną ponownie pobrane z API zewnętrznego.

Podstawowy URL API

<http://localhost:8080/api/posts>

Endpointy

1. Pobierz wszystkie posty

GET /api/posts

- **Opis:** Pobiera wszystkie posty zapisane w pamięci.
- **Odpowiedź:** Zwraca listę wszystkich postów.

Przykładowa odpowiedź:

Json:

```
[
  {
    "userId": 1,
    "id": 1,
    "title": "Tytuł posta 1",
    "body": "To jest treść posta 1."
  },
  {
    "userId": 1,
    "id": 2,
    "title": "Tytuł posta 2",
    "body": "To jest treść posta 2."
  },
]
```

2. Pobierz post po ID

GET /api/posts/{id}

- **Opis:** Pobiera pojedynczy post na podstawie jego ID.
- **Parametr ścieżki:**
 - id (integer): ID posta do pobrania.
- **Odpowiedź:** Zwraca post o podanym ID.
- **Błąd:** Jeśli post o podanym ID nie istnieje, zwraca błąd 500 “Nie znaleziono posta o id: {id}”.

Przykładowa odpowiedź:

```
json
{
  "userId": 1,
  "id": 1,
  "title": "Tytuł posta 1",
  "body": "To jest treść posta 1."
}
```

Przykładowy błąd:

```
json
{
  "timestamp": "2025-01-26T22:15:53.802+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "Nie znaleziono posta o id: 1",
  "path": "/api/posts/1"
}
```

3. Dodaj nowy post

POST /api/posts

- **Opis:** Dodaje nowy post z określonym ID. Jeśli ID jest już zajęte, zwraca błąd.
- **Treść żądania:**
 - id (integer): ID posta do dodania (musi być unikalne).
 - title (string): Tytuł posta.
 - body (string): Treść posta.

Przykładowe żądanie:

```
json
{
  "userId": 1,
  "id": 1,
  "title": "Nowy Tytuł Posta",
  "body": "To jest treść nowego posta."
}
```

Przykładowa odpowiedź:

```
json
{
  "userId": 1,
  "id": 1,
  "title": "Tytuł posta 1",
  "body": "To jest treść posta 1."
}
```

Przykładowy błąd:

```
json
{
  "timestamp": "2025-01-26T22:15:53.802+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "Post o id: 1 już istnieje",
  "path": "/api/posts/1"
}
```

4. Zaktualizuj istniejący post

PUT /api/posts/{id}

- **Opis:** Aktualizuje tytuł i treść istniejącego posta na podstawie jego ID.
- **Parametr ścieżki:**
 - id (integer): ID posta do zaktualizowania.
- **Treść żądania:**
 - title (string): Nowy tytuł posta.
 - body (string): Nowa treść posta.

Przykładowe żądanie:

```
json
{
  "title": "Zaktualizowany Tytuł Posta",
  "body": "Zaktualizowana treść posta."
}
```

Przykładowa odpowiedź:

```
json
{
  "userId": 1,
  "id": 1,
  "title": "Zaktualizowany tytuł",
  "body": "Zaktualizowana treść posta."
}
```

Przykładowy błąd:

```
json
{
  "timestamp": "2025-01-26T22:50:00.000+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "Nie znaleziono posta o id: 1",
  "path": "/api/posts/1"
}
```

5. Usuń post

DELETE /api/posts/{id}

- **Opis:** Usuwa post na podstawie jego ID.
- **Parametr ścieżki:**
 - id (integer): ID posta do usunięcia.
- **Odpowiedź:** Zwraca No Content, jeśli post został pomyślnie usunięty.

Przykładowy błąd:

```
json
{
  "timestamp": "2025-01-26T22:50:00.000+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "Nie udało się usunąć posta o id: 1",
  "path": "/api/posts/1"
}
```