

Dokumentacja API Supabase dla bazy danych produktów

Spis treści

1. [Wprowadzenie](#)
2. [Zarządzanie produktami](#)
3. [Zarządzanie kategoriami](#)
4. [Zarządzanie historią cen](#)
5. [Wyszukiwanie i filtrowanie](#)
6. [Statystyki i raporty](#)
7. [Pobieranie szczegółów produktu](#)

Wprowadzenie

Poniższa dokumentacja zawiera opis funkcji API dla bazy danych produktów, ich kategorii i historii cen, dostępnych przez Supabase. Wszystkie zapytania należy kierować na endpoint:

```
https://[twój-projekt].supabase.co/rest/v1/rpc/[nazwa_funkcji]
```

Wymagane nagłówki:

```
apikey: [twój-klucz-api]
Authorization: Bearer [twój-klucz-api]
Content-Type: application/json
```

Zarządzanie produktami

1. Dodawanie nowego produktu

Funkcja: `add_product`

Dodaje nowy produkt do bazy danych.

Parametry:

- `title` (TEXT) - tytuł produktu (wymagany)
- `description` (TEXT) - opis produktu
- `price` (NUMERIC) - cena produktu
- `url` (TEXT) - URL do strony produktu
- `product_source` (VARCHAR) - źródło produktu (np. "web", "store")
- `currency` (VARCHAR) - waluta (np. "PLN", "EUR")

- **rating** (NUMERIC) - ocena produktu (np. 4.5)
- **review_count** (INTEGER) - liczba recenzji
- **img_url** (TEXT) - URL do obrazka produktu

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-projekt].supabase.co/rest/v1/rpc/add_product' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "title": "Laptop Dell XPS 13",
  "description": "Laptop z procesorem Intel Core i7, 16GB RAM, 512GB
SSD",
  "price": 4999.99,
  "url": "https://example.com/dell-xps-13",
  "product_source": "web",
  "currency": "PLN",
  "rating": 4.7,
  "review_count": 245,
  "img_url": "https://example.com/images/dell-xps-13.jpg"
}'
```

Odpowiedź:

Identyfikator nowo dodanego produktu (INTEGER).

2. Aktualizacja produktu

Funkcja: **update_product**

Aktualizuje istniejący produkt.

Parametry:

- **p_product_id** (INTEGER) - identyfikator produktu (wymagany)
- **p_title** (TEXT) - nowy tytuł produktu (opcjonalny)
- **p_description** (TEXT) - nowy opis (opcjonalny)
- **p_price** (NUMERIC) - nowa cena (opcjonalna)
- **p_url** (TEXT) - nowy URL (opcjonalny)
- **p_product_source** (VARCHAR) - nowe źródło (opcjonalne)
- **p_currency** (VARCHAR) - nowa waluta (opcjonalna)
- **p_rating** (NUMERIC) - nowa ocena (opcjonalna)
- **p_review_count** (INTEGER) - nowa liczba recenzji (opcjonalna)
- **p_img_url** (TEXT) - nowy URL obrazka (opcjonalny)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/update_product' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_product_id": 1,
  "p_price": 4899.99,
  "p_description": "Laptop z procesorem Intel Core i7, 16GB RAM, 512GB
SSD - promocja!"
}'
```

Odpowiedź:

Wartość **true** jeśli aktualizacja się powiodła, **false** jeśli nie znaleziono produktu.

Zarządzanie kategoriami

3. Dodawanie nowej kategorii

Funkcja: **add_category**

Dodaje nową kategorię produktów.

Parametry:

- **p_category_name** (TEXT) - nazwa kategorii (wymagana)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-projekt].supabase.co/rest/v1/rpc/add_category' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_category_name": "Laptopy premium"
}'
```

Odpowiedź:

Identyfikator nowo dodanej kategorii (UUID).

4. Przypisanie produktu do kategorii

Funkcja: `assign_product_to_category`

Przypisuje produkt do kategorii. Jeśli kategoria nie istnieje, zostanie utworzona.

Parametry:

- `p_product_id` (INTEGER) - identyfikator produktu (wymagany)
- `p_category_name` (TEXT) - nazwa kategorii (wymagana)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/assign_product_to_category' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_product_id": 1,
  "p_category_name": "Laptopy premium"
}'
```

Odpowiedź:

Wartość `true` jeśli przypisanie się powiodło, `false` jeśli produkt już był przypisany do tej kategorii.

Zarządzanie historią cen

5. Dodawanie wpisu historii cen

Funkcja: `add_price_history`

Dodaje nowy wpis do historii cen produktu.

Parametry:

- `p_product_id` (INTEGER) - identyfikator produktu (wymagany)
- `p_price` (NUMERIC) - cena produktu (wymagana)
- `p_history_id` (VARCHAR) - opcjonalny identyfikator wpisu historii; jeśli nie podano, zostanie wygenerowany automatycznie w formacie YYYYMMDD-product_id

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/add_price_history' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
```

```
-d '{
  "p_product_id": 1,
  "p_price": 4999.99,
  "p_history_id": "20240401-1"
}'
```

Odpowiedź:

Identyfikator wpisu historii cen (VARCHAR).

Wyszukiwanie i filtrowanie

6. Wyszukiwanie produktów

Funkcja: `search_products`

Wyszukuje produkty według różnych kryteriów.

Parametry:

- `p_search_term` (TEXT) - wyszukiwana fraza w tytule lub opisie (opcjonalna)
- `p_min_price` (NUMERIC) - minimalna cena (opcjonalna)
- `p_max_price` (NUMERIC) - maksymalna cena (opcjonalna)
- `p_min_rating` (NUMERIC) - minimalna ocena (opcjonalna)
- `p_category` (TEXT) - nazwa kategorii (opcjonalna)
- `p_limit` (INTEGER) - limit wyników, domyślnie 50 (opcjonalny)
- `p_offset` (INTEGER) - przesunięcie wyników, domyślnie 0 (opcjonalne)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/search_products' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_search_term": "laptop",
  "p_min_price": 3000,
  "p_max_price": 6000,
  "p_min_rating": 4.5,
  "p_limit": 10
}'
```

Odpowiedź:

Tablica obiektów JSON zawierających dane produktów wraz z przypisanymi kategoriami.

Statystyki i raporty

7. Statystyki cenowe produktu

Funkcja: `get_product_price_stats`

Zwraca statystyki cenowe dla konkretnego produktu.

Parametry:

- `p_product_id` (INTEGER) - identyfikator produktu (wymagany)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-  
projekt].supabase.co/rest/v1/rpc/get_product_price_stats' \  
-H "apikey: [twój-klucz-api]" \  
-H "Authorization: Bearer [twój-klucz-api]" \  
-H "Content-Type: application/json" \  
-d '{  
  "p_product_id": 1  
}'
```

Odpowiedź:

Obiekt JSON zawierający statystyki cenowe:

- `product_id` - identyfikator produktu
- `title` - tytuł produktu
- `current_price` - aktualna cena
- `min_price` - minimalna cena w historii
- `max_price` - maksymalna cena w historii
- `avg_price` - średnia cena
- `price_count` - liczba wpisów w historii cen
- `first_tracked` - najwcześniejszy wpis historii
- `last_tracked` - najnowszy wpis historii

8. Statystyki kategorii

Funkcja: `get_category_stats`

Zwraca statystyki dla wszystkich kategorii.

Parametry:

Brak parametrów.

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-  
projekt].supabase.co/rest/v1/rpc/get_category_stats' \  
-H "apikey: [twój-klucz-api]" \  
-H "Authorization: Bearer [twój-klucz-api]" \  
-H "Content-Type: application/json" \  
-d '{}'
```

Odpowiedź:

Tablica obiektów JSON zawierających statystyki dla każdej kategorii:

- **category_name** - nazwa kategorii
- **product_count** - liczba produktów w kategorii
- **avg_price** - średnia cena produktów
- **min_price** - minimalna cena produktu
- **max_price** - maksymalna cena produktu
- **avg_rating** - średnia ocena produktów

Pobieranie szczegółów produktu

9. Szczegóły produktu

Funkcja: **get_product_details**

Pobiera pełne szczegóły produktu wraz z historią cen i kategoriami.

Parametry:

- **p_product_id** (INTEGER) - identyfikator produktu (wymagany)

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-  
projekt].supabase.co/rest/v1/rpc/get_product_details' \  
-H "apikey: [twój-klucz-api]" \  
-H "Authorization: Bearer [twój-klucz-api]" \  
-H "Content-Type: application/json" \  
-d '{  
  "p_product_id": 1  
'
```

Odpowiedź:

Obiekt JSON zawierający:

- **product** - szczegóły produktu
- **price_history** - tablica wpisów historii cen
- **categories** - tablica nazw kategorii przypisanych do produktu

Przykłady użycia w różnych językach

Python

```
import requests

url = "https://[twój-projekt].supabase.co/rest/v1/rpc/search_products"
headers = {
    "apikey": "[twój-klucz-api]",
    "Authorization": "Bearer [twój-klucz-api]",
    "Content-Type": "application/json"
}
data = {
    "p_search_term": "laptop",
    "p_min_price": 3000
}

response = requests.post(url, headers=headers, json=data)
results = response.json()
print(results)
```

JavaScript

```
fetch('https://[twój-projekt].supabase.co/rest/v1/rpc/search_products', {
  method: 'POST',
  headers: {
    'apikey': '[twój-klucz-api]',
    'Authorization': `Bearer [twój-klucz-api]`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    p_search_term: 'laptop',
    p_min_price: 3000
  })
})
.then(response => response.json())
.then(data => console.log(data));
```

Java


```
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

HttpClient client = HttpClient.newHttpClient();
String json = "{\"p_search_term\": \"laptop\", \"p_min_price\": 3000}";

HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create("https://[twój-
projekt].supabase.co/rest/v1/rpc/search_products"))
    .header("apikey", "[twój-klucz-api]")
    .header("Authorization", "Bearer [twój-klucz-api]")
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(json))
    .build();

HttpResponse<String> response = client.send(request,
HttpResponse.BodyHandlers.ofString());
System.out.println(response.body());
```