

Dokumentacja API Supabase dla bazy danych produktów

Data aktualizacji: 2025-04-23

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](#)
- [8. Nowe funkcje zastępujące zapytania JPA](#)
- [9. Przykłady użycia w różnych językach](#)

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

Nowe funkcje zastępujące zapytania JPA

Poniższe funkcje zostały specjalnie zaimplementowane, aby zastąpić zapytania JPA/Hibernate używane wcześniej w aplikacji.

10. Pobieranie wszystkich produktów z paginacją

Funkcja: `find_all_products`

Zastępuje zapytanie JPA: `@Query("SELECT p FROM Product p") Page<Product> findAllProducts(Pageable pageable);`

Parametry:

- **p_page_size** (INTEGER) - liczba produktów na stronę, domyślnie 20
- **p_page_number** (INTEGER) - numer strony (licząc od 0), domyślnie 0

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/find_all_products' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_page_size": 20,
  "p_page_number": 0
}'
```

Odpowiedź: Tablica obiektów produktów z danej strony.

11. Pobieranie całkowitej liczby produktów

Funkcja: `count_all_products`

Pomocnicza funkcja zwracająca całkowitą liczbę produktów (dla paginacji).

Parametry: Brak parametrów.

Przykładowe zapytanie:

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

Odpowiedź: Liczba całkowita reprezentująca ilość produktów w bazie danych.

12. Wyszukiwanie produktów według wielu kategorii

Funkcja: `find_products_by_category_names`

Zastępuje zapytanie JPA: `@Query("SELECT DISTINCT p FROM Product p JOIN ProductCategory pc ON p.productId = pc.id.productId JOIN Category c ON pc.id.categoryId = c.categoryId WHERE c.categoryName IN :categories") Page<Product>`
`findProductsByCategoryNames(@Param("categories") List<String> categoryNames, Pageable pageable);`

Parametry:

- `p_categories` (TEXT[]) - tablica nazw kategorii
- `p_page_size` (INTEGER) - liczba produktów na stronę, domyślnie 20
- `p_page_number` (INTEGER) - numer strony (licząc od 0), domyślnie 0

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/find_products_by_category_names' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_categories": ["Laptopy", "Elektronika"],
  "p_page_size": 20,
  "p_page_number": 0
}'
```

Odpowiedź: Tablica obiektów produktów należących do podanych kategorii.

13. Liczba produktów w kategoriach

Funkcja: `count_products_by_categories`

Pomocnicza funkcja zwracająca liczbę produktów w podanych kategoriach (dla paginacji).

Parametry:

- `p_categories` (TEXT[]) - tablica nazw kategorii

Przykładowe zapytanie:

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

Odpowiedź: Liczba całkowita reprezentująca ilość produktów w podanych kategoriach.

14. Wyszukiwanie produktów po fragmencie tytułu

Funkcja: `search_products_by_key`

Zastępuje zapytanie JPA: `@Query("SELECT p FROM Product p WHERE LOWER(p.title) LIKE LOWER(CONCAT('%', :query, '%'))") Page<Product> searchProductsByKey(@Param("query") String query, Pageable pageable);`

Parametry:

- `p_query` (TEXT) - fraza do wyszukania w tytule
- `p_page_size` (INTEGER) - liczba produktów na stronę, domyślnie 20
- `p_page_number` (INTEGER) - numer strony (licząc od 0), domyślnie 0

Przykładowe zapytanie:

```
curl -X POST 'https://[twój-
projekt].supabase.co/rest/v1/rpc/search_products_by_key' \
-H "apikey: [twój-klucz-api]" \
-H "Authorization: Bearer [twój-klucz-api]" \
-H "Content-Type: application/json" \
-d '{
  "p_query": "laptop",
  "p_page_size": 20,
  "p_page_number": 0
}'
```

Odpowiedź: Tablica obiektów produktów z frazą w tytule.

15. Liczba produktów z frazą w tytule

Funkcja: `count_products_by_key`

Pomocnicza funkcja zwracająca liczbę produktów z frazą w tytule (dla paginacji).

Parametry:

- `p_query` (TEXT) - fraza do wyszukania w tytule

Przykładowe zapytanie:

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

Odpowiedź: Liczba całkowita reprezentująca ilość produktów z frazą w tytule.

Przykłady użycia w różnych językach

Python

```
import requests

# Pobieranie wszystkich produktów (pierwsza strona, 20 elementów)
url = "https://[twój-projekt].supabase.co/rest/v1/rpc/find_all_products"
headers = {
    "apikey": "[twój-klucz-api]",
    "Authorization": "Bearer [twój-klucz-api]",
    "Content-Type": "application/json"
}
data = {
    "p_page_size": 20,
    "p_page_number": 0
}
response = requests.post(url, headers=headers, json=data)
products = response.json()
print(f"Liczba pobranych produktów: {len(products)}")

# Pobieranie liczby wszystkich produktów
url = "https://[twój-projekt].supabase.co/rest/v1/rpc/count_all_products"
response = requests.post(url, headers=headers, json={})
total_count = response.json()
print(f"Całkowita liczba produktów: {total_count}")
```

```
# Wyszukiwanie produktów w kategoriach "Laptopy" i "Elektronika"
url = "https://[twój-projekt].supabase.co/rest/v1/rpc/find_products_by_category_names"
data = {
  "p_categories": ["Laptopy", "Elektronika"],
  "p_page_size": 20,
  "p_page_number": 0
}
response = requests.post(url, headers=headers, json=data)
category_products = response.json()
print(f"Liczba produktów w kategoriach: {len(category_products)}")

# Wyszukiwanie produktów z frazą "laptop" w tytule
url = "https://[twój-projekt].supabase.co/rest/v1/rpc/search_products_by_key"
data = {
  "p_query": "laptop",
  "p_page_size": 20,
  "p_page_number": 0
}
response = requests.post(url, headers=headers, json=data)
search_results = response.json()
print(f"Liczba wyników wyszukiwania: {len(search_results)}")
```

JavaScript

```
// Pobieranie wszystkich produktów (pierwsza strona, 20 elementów)
fetch('https://[twój-projekt].supabase.co/rest/v1/rpc/find_all_products', {
  method: 'POST',
  headers: {
    'apikey': '[twój-klucz-api]',
    'Authorization': `Bearer [twój-klucz-api]`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    p_page_size: 20,
    p_page_number: 0
  })
})
.then(response => response.json())
.then(products => console.log(`Liczba pobranych produktów: ${products.length}`));

// Pobieranie liczby wszystkich produktów
fetch('https://[twój-projekt].supabase.co/rest/v1/rpc/count_all_products', {
  method: 'POST',
  headers: {
    'apikey': '[twój-klucz-api]',
    'Authorization': `Bearer [twój-klucz-api]`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    p_page_size: 20,
    p_page_number: 0
  })
})
.then(response => response.json())
.then(count => console.log(`Liczba wszystkich produktów: ${count}`));
```

```
    },
    body: JSON.stringify({})
  })
  .then(response => response.json())
  .then(count => console.log(`Całkowita liczba produktów: ${count}`));

// Wyszukiwanie produktów w kategoriach "Laptopy" i "Elektronika"
fetch('https://[twój-projekt].supabase.co/rest/v1/rpc/find_products_by_category_names', {
  method: 'POST',
  headers: {
    'apikey': '[twój-klucz-api]',
    'Authorization': `Bearer [twój-klucz-api]`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    p_categories: ["Laptopy", "Elektronika"],
    p_page_size: 20,
    p_page_number: 0
  })
})
  .then(response => response.json())
  .then(products => console.log(`Liczba produktów w kategoriach: ${products.length}`));

// Wyszukiwanie produktów z frazą "laptop" w tytule
fetch('https://[twój-projekt].supabase.co/rest/v1/rpc/search_products_by_key', {
  method: 'POST',
  headers: {
    'apikey': '[twój-klucz-api]',
    'Authorization': `Bearer [twój-klucz-api]`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    p_query: "laptop",
    p_page_size: 20,
    p_page_number: 0
  })
})
  .then(response => response.json())
  .then(products => console.log(`Liczba wyników wyszukiwania: ${products.length}`));
```

Java

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

```
HttpClient client = HttpClient.newHttpClient();

// Pobieranie wszystkich produktów (pierwsza strona, 20 elementów)
String findAllJson = "{\"p_page_size\": 20, \"p_page_number\": 0}";
HttpRequest findAllRequest = HttpRequest.newBuilder()
    .uri(URI.create("https://[twój-projekt].supabase.co/rest/v1/rpc/find_all_products"))
    .header("apikey", "[twój-klucz-api]")
    .header("Authorization", "Bearer [twój-klucz-api]")
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(findAllJson))
    .build();
HttpResponse<String> findAllResponse = client.send(findAllRequest,
    HttpResponse.BodyHandlers.ofString());
System.out.println("Odpowiedź find_all_products: " +
    findAllResponse.body());

// Wyszukiwanie produktów w kategoriach "Laptopy" i "Elektronika"
String categoriesJson = "{\"p_categories\": [\"Laptopy\", \"Elektronika\"], \"p_page_size\": 20, \"p_page_number\": 0}";
HttpRequest categoriesRequest = HttpRequest.newBuilder()
    .uri(URI.create("https://[twój-projekt].supabase.co/rest/v1/rpc/find_products_by_category_names"))
    .header("apikey", "[twój-klucz-api]")
    .header("Authorization", "Bearer [twój-klucz-api]")
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(categoriesJson))
    .build();
HttpResponse<String> categoriesResponse = client.send(categoriesRequest,
    HttpResponse.BodyHandlers.ofString());
System.out.println("Odpowiedź find_products_by_category_names: " +
    categoriesResponse.body());

// Wyszukiwanie produktów z frazą "laptop" w tytule
String searchJson = "{\"p_query\": \"laptop\", \"p_page_size\": 20, \"p_page_number\": 0}";
HttpRequest searchRequest = HttpRequest.newBuilder()
    .uri(URI.create("https://[twój-projekt].supabase.co/rest/v1/rpc/search_products_by_key"))
    .header("apikey", "[twój-klucz-api]")
    .header("Authorization", "Bearer [twój-klucz-api]")
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(searchJson))
    .build();
HttpResponse<String> searchResponse = client.send(searchRequest,
    HttpResponse.BodyHandlers.ofString());
System.out.println("Odpowiedź search_products_by_key: " +
    searchResponse.body());
```