



Przykład struktury JSON dla nowego produktu (API Shoper)

Dodawanie grafiki produktu (endpoint /webapi2/gfx)

Zanim utworzysz produkt, najpierw dodaj obraz produktu przez API Shopera. W tym celu użyj endpointu `/webapi2/gfx` (w nowszym REST API odpowiada mu zasób `product-images`). W żądaniu **POST** przekaż plik graficzny (np. w formie ciągu Base64) wraz z nazwą pliku. Przykładowo struktura body powinna zawierać obiekt `gfx` z polami:

- **gfx.file** – nazwa pliku (np. `"pikachu.jpg"`).
- **gfx.content** – zawartość obrazka zakodowana w Base64 ¹.

Po poprawnym dodaniu obrazka API zwróci `gfx_id` – unikalne ID grafiki. Zanotuj je, aby użyć przy tworzeniu produktu.

Tworzenie nowego produktu – kluczowe pola JSON

Kolejnym krokiem jest **utworzenie produktu** poprzez endpoint `/webapi/rest/products` metodą **POST**. Poniżej omawiamy najważniejsze pola JSON niezbędne dla sklepu z kartami Pokémon TCG:

- **category_id** – ID kategorii, do której produkt ma należeć. Należy podać właściwy identyfikator istniejącej kategorii. Np. dla kategorii *Surging Sparks* ID to **58** ².
- **code** – Kod produktu (wewnętrzny kod lub symbol produktu).
- **stock.code** – Kod magazynowy (SKU) konkretnej pozycji magazynowej produktu ³. Pozwala odróżnić warianty lub używać kodów magazynowych niezależnie od głównego kodu produktu.
- **stock.price** – Cena **brutto** produktu (liczba zmiennoprzecinkowa). Jest to podstawowa cena jednostkowa widoczna dla klienta (np. `25.99` oznacza 25,99 zł) ⁴.
- **stock.stock** – Stan magazynowy, czyli liczba sztuk produktu w magazynie (np. `30.0` oznacza 30 sztuk).
- **stock.availability_id** – ID dostępności produktu, określający komunikat typu „dostępny”, „brak” itp. Np. **5** oznacza status *duża ilość* (stosowany gdy produkt jest w dużej ilości na stanie) ⁵. Wartość ta wpływa na wyświetlany komunikat o dostępności.
- **stock.delivery_id** – ID czasu dostawy dla produktu. Określa zdefiniowany w sklepie komunikat, za ile czasu towar jest wysyłany (np. ID 1 może oznaczać „Wysyłka w 24h”).
- **gfx_id** – ID grafiki produktu (pobrany we wcześniejszym kroku). W strukturze produktu jest on umieszczany zwykle wewnątrz obiektu **stock** jako **stock.gfx_id**, wskazując główne zdjęcie przypisane do tego produktu ⁶.
- **translations** – Tłumaczenia i nazwy produktu dla poszczególnych języków. Jest to obiekt, gdzie kluczem jest kod języka (np. `"pl_PL"`). Wewnątrz należy podać:
- **name** – Nazwa produktu, np. „*Pikachu EX*”.
- **short_description** – Krótki opis produktu (np. jednozdaniowy opis wyróżniający kartę).
- **description** – Pełny opis produktu (może zawierać tekst sformatowany HTML, szczegóły karty, stan itp.).

- **active** – Flaga aktywności dla danego języka (bool). Ustaw **true**, aby produkt był aktywny (widoczny) w sklepie ⁴.
- **attributes** – Lista atrybutów produktu (np. cechy karty Pokémon). Każdy atrybut dodajemy jako obiekt z kluczem **attribute_id** (ID atrybutu) oraz **option_id** (wybraną opcję tego atrybutu).
- Przykładowo:
- *Rzadkość* (**attribute_id: 38**) – opcja **Rare** (*Rzadka karta*, **option_id: 117**) ⁷,
- *Typ karty* (**attribute_id: 39**) – opcja **Pokémon EX** (**option_id: 118**) ⁸,
- *Typ Pokémona* (np. energia **Lightning**, **attribute_id: 63**, **option_id: 128**) ⁹.
- **newproduct** – Flaga boolean oznaczająca produkt *nowy* w ofercie. Ustaw **true** dla nowo dodanych produktów, jeśli chcesz wyróżnić je jako nowości ¹⁰.
- **bestseller** – Flaga boolean oznaczająca *bestseller*. Ustaw **true**, jeśli produkt ma być oznaczony jako bestseller w sklepie ¹¹. (Obie flagi można ustawić jednocześnie wedle potrzeb.)
- **unit_id** – (Opcjonalne) ID jednostki miary produktu. Jeśli sklep korzysta z jednostek (sztuki, opakowania itp.), podaj odpowiednie ID (np. **1** zazwyczaj oznacza sztuki).
- **currency_id** – (Opcjonalne) ID waluty, w której podana jest cena. Jeśli pomijesz, zostanie użyta domyślna waluta sklepu (np. PLN z ID **1**).

Uwaga: Pola wymagane to m.in. **price** (cena) – należy go podać, inaczej API zwróci błąd, oraz **translations.name**. Inne pola, które nie mają wartości domyślnych ani nie są opcjonalne (wg dokumentacji), również muszą zostać uwzględnione. ID liczbowych (jak **category_id**, **attribute_id**, **option_id**, **gfx_id** itp.) należy podawać jako liczby, nie ciągi znaków (bez cudzysłów), natomiast teksty (nazwy, opisy, kody) jako stringi ujęte w cudzysłów.

Przykładowe zapytanie POST – utworzenie produktu

Poniżej znajduje się **przykładowy JSON** użyty w zapytaniu **POST** do utworzenia nowego produktu (np. karty *Pikachu EX*) poprzez API Shoper. Zawiera on wszystkie omówione wyżej kluczowe pola:

```

POST /webapi/rest/products
Content-Type: application/json

{
  "category_id": 58,
  "code": "PIKA-EX-001",
  "unit_id": 1,
  "currency_id": 1,
  "stock": {
    "code": "SKU-PIKA-001",
    "price": 25.99,
    "stock": 30.0,
    "availability_id": 5,
    "delivery_id": 1,
    "gfx_id": 12345      // ID obrazka z poprzedniego kroku (/webapi2/gfx)
  },
  "attributes": [
    { "attribute_id": 38, "option_id": 117 }, // Rzadkość: Rare
    { "attribute_id": 39, "option_id": 118 }, // Typ karty: Pokémon EX
    { "attribute_id": 63, "option_id": 128 }   // Typ Pokémona: Lightning
  ],
  "translations": {
    "name": "Pikachu EX"
  }
}
  
```

```

    "pl_PL": {
        "name": "Pikachu EX",
        "short_description": "Karta Pok  mon Pikachu EX z edycji Surging Sparks.",
        "description": "Pikachu EX - rzadka karta z dodatku Surging Sparks. Stan idealny, w folii ochronnej.",
        "active": true
    },
    "newproduct": true,
    "bestseller": false
}

```

W powy szym przyk adzie:

- Produkt zostanie dodany do kategorii o ID 58 (*Surging Sparks*).
- Otrzymuje kody: *PIKA-EX-001* (kod produktu) oraz *SKU-PIKA-001* (kod magazynowy).
- Cena brutto wynosi 25,99  , stan magazynowy to 30 sztuk, a dost pno st ustawiona jest na ID 5 (*du a ilo s * [5](#)).
- **gfx_id** [12345](#) wskazuje wcze ej dodany obrazek karty (ten ID nale y podmieni  na faktyczny identyfikator zwr cony przez API po dodaniu grafiki).
- Ustawiono atrybuty: **Rzadko : Rare**, **Typ karty: Pok  mon EX**, **Typ Pok  mona: Lightning** (zgodnie z odpowiednimi ID opcji w systemie [7](#) [8](#)).
- W sekcji **translations** podano polsk  nazw  oraz opisy produktu.
- Flaga **newproduct=true** oznacza,  e produkt b dzie wyr zniony jako nowo , a **bestseller=false** (produkt nie b dzie oznaczony jako bestseller).

Po wyslaniu takiego zapytania API powinno utworzy  nowy produkt i zwr ci m.in. **product_id** nowego produktu. W razie potrzeby dalszej aktualizacji stanu magazynowego lub innych danych, mo na u y  odpowiednich endpointów (np. [/webapi/rest/product-stocks/](#) do edycji stanu, je li nie zrobiono tego od razu). Powy sza struktura JSON stanowi kompletny szkielet danych niezb dnych do dodania karty Pok  mon TCG przez API Shoper, zgodnie z dokumentacj  i wymaganiami systemu [4](#).

[1](#) Producers: insert - Shoper Developers

<https://developers.shoper.pl/developers/api/resources/producers/insert>

[2](#) [5](#) [7](#) [8](#) [9](#) ids_dump.json

file:///file_00000000c60471f4b3327c5032f67402

[3](#) [11](#) Products - Shoper Developers

<https://developers.shoper.pl/developers/api/resources/products>

[4](#) Shoper i REST API :: 4programmers.net

https://4programmers.net/Forum/C_i_.NET/322855-shoper_i_rest_api

[6](#) Product Stock: list - Shoper Developers

<https://developers.shoper.pl/developers/api/resources/product-stocks/list>

[10](#) Products details context - Shoper Developers

<https://developers.shoper.pl/developers/snippets/contexts/product-details>