Integracja z systemami zewn?trznymi

This page contains infromations and instructions on how to get data from AMPER for use in external services (like e-commerce site).

- API zewnętrzne – integracja sklepu internetowego

API zewn?trzne – integracja sklepu internetowego

API zewnętrzne AMPER (`/external/v1/`) udostępnia do odczytu katalog produktów, kategorie, zdjęcia, stany magazynowe, promocje i rodzaje wysyłki. Dane są zawsze widziane oczami jednego klienta (kontrahenta) – z jego cenami, asortymentem i promocjami. Typowe zastosowania: zasilenie sklepu internetowego, porównywarki cenowej albo wygenerowanie pliku XML.

Przykładowy projekt: generator pliku XML dla CENEO.PL.

Czego potrzebujesz

1. **Adres serwera AMPER WS** – np. `https://amper-xyz-ws.ampliapps.com` (w przykładach: `{amper_ws_url}`).
2. **Klucz API** – identyfikator w formacie UUID (w przykładach: `{amper_api_key}`).

Oba otrzymasz od działu IT firmy, z którą się integrujesz. Klucz generuje administrator w panelu AMPER i przypisuje go do konkretnego konta klienta.

Uwierzytelnienie

Klucz przekazujesz w parametrze `api_key` w adresie każdego zapytania:

```
GET https://{amper_ws_url}/external/v1/products/?api_key={amper_api_key}
```

- API obsługuje wyłącznie metodę **GET**.
- Klucz jest przypisany do konta klienta: API zwraca produkty dostępne dla tego klienta, jego ceny i jego promocje – tak, jak widzi je w sklepie B2B.
- Klucz ma datę ważności i administrator może go wyłączyć. Wygasły lub wyłączony klucz daje odpowiedź `401`.
- Klucz daje dostęp do danych handlowych klienta. Przechowuj go po stronie serwera – nigdy w kodzie strony www ani aplikacji mobilnej.

Dost?pne zasoby

Każdy zasób ma listę (`/external/v1/<zasób>/`) i odczyt pojedynczego obiektu (`/external/v1/<zasób>/{id}/`).

Adres	Zawartość
`/external/v1/products/`	produkty dostępne dla klienta, z cenami
`/external/v1/categories/`	kategorie produktów
`/external/v1/product-categories/`	przypisania produktów do kategorii
`/external/v1/product-files/`	zdjęcia produktów
`/external/v1/stocks/`	stany magazynowe produktów
`/external/v1/promotions/`	promocje klienta – opis w artykule Promocje – integracja z systemami zewnętrznymi
`/external/v1/shipment-types/`	rodzaje wysyłki

Przydatne filtry:

- `/external/v1/products/?ids=101,205,317` – tylko wskazane produkty,
- `/external/v1/products/?category_id=12` – produkty z kategorii,
- `/external/v1/product-files/?product=101` albo `?product_external_id=SKU-ABC-001` – zdjęcia jednego produktu.

Paginacja

- **Bez parametrów** odpowiedź zawiera całą listę jako tablicę JSON.
- **Z parametrami** `limit` i `offset` odpowiedź jest stroną w kopercie:

```
{
  "count": 1817,
  "next": "https://{amper_ws_url}/external/v1/products/?api_key=...&limit=500&offset=500",
  "previous": null,
  "results": [ ... ]
}
```

Przy dużym katalogu pobieraj dane stronami (np. `limit=500`), aż `next` będzie równe `null`.

Limity

- **Najwyżej 3 jednoczesne zapytania na klucz.** Kolejne dostają odpowiedź `429 Too Many Requests` z nagłówkiem `Retry-After: 1` – odczekaj wskazaną liczbę sekund i ponów

zapytanie. Strony pobieraj po kolei albo najwyżej dwie, trzy naraz.
- **Około 800 zapytań na minutę z jednego adresu IP.**
- Dane katalogowe (produkty, kategorie, zdjęcia) zmieniają się rzadko – wystarczy je odświeżać co kilka godzin. Częściej odświeżaj tylko stany i ceny. Nie pobieraj całego katalogu co kilka minut.

Odpowiedzi

Produkt


```
{
  "id": 1037,
  "name": "BAHAMA zapach samochodowy NASZYJNIK Wild Hibiskus /1szt",
  "friendly_name": "Zapach samochodowy Bahama Wild Hibiskus",
  "short_code": "850267",
  "description": "Zapachy inspirowane wakacjami oraz trendami konsumenckimi.",
  "short_description": "",
  "ean": "5904000850267",
  "sku": "850267",
  "cn_code": "33074900",
  "vat": 23,
  "default_unit_of_measure": "szt.",
  "weight": "0.20",
  "default_price": 25.43,
  "currency_code": "PLN",
  "product_external_id": "850267",
  "list_price": 29.90,
  "cumulative_unit_of_measure": "karton",
  "cumulative_converter": "24.00",
  "can_be_split": true,
  "cumulative_unit_ratio_splitter": "6.00",
  "unit_roundup": false,
  "capacity": null,
  "stock_availability_customer": 1,
  "brand": "Bahama & Co.",
  "manufacturer": "Bahama",
  "attributes": [
    {
      "id": 4,
```

```
      "slug": "zapach",
      "name": "Zapach",
      "values": [
        { "id": 31, "name": "Hibiskus", "slug": "hibiskus-31", "selected": true, "icon": null
}
      ],
      "color": null,
      "show_on_tile": false
    }
  ],
  "deposit_pack_multiplier": null,
  "deposit_pack_price": null,
  "deposit_pack_type": null
}
```


Pole	Opis
`id`	ID produktu w AMPER
`name`, `friendly_name`	nazwa pełna i nazwa handlowa (krótsza, „przyjazna”)
`short_code`, `sku`, `ean`	kod produktu, SKU, kod EAN
`description`, `short_description`	opis i krótki opis
`cn_code`	kod CN (Nomenklatura scalona)
`vat`	stawka VAT w procentach
`default_unit_of_measure`	podstawowa jednostka miary
`weight`	waga
`default_price`	**cena netto dla klienta przypisanego do klucza** – uwzględnia jego cennik i rabaty
`currency_code`	waluta cen klienta
`list_price`	cena katalogowa netto
`product_external_id`	ID produktu w systemie ERP
`cumulative_unit_of_measure`, `cumulative_converter`	jednostka zbiorcza (np. karton) i liczba jednostek podstawowych w niej
`can_be_split`, `cumulative_unit_ratio_splitter`	czy jednostkę zbiorczą można dzielić i na ile części
`unit_roundup`	czy zamawiana ilość jest zaokrąglana w górę do kroku jednostki
`capacity`	pojemność

Pole	Opis
`stock_availability_customer`	dostępność dla klienta: rzeczywista ilość (stan minus rezerwacje), gdy klient ma włączoną opcję *Zwracaj rzeczywistą wartość stanów w API i ofertach*; w przeciwnym razie `1` (jest na stanie) lub `0`
`brand`, `manufacturer`	marka i producent
`attributes`	atrybuty produktu oznaczone w panelu jako widoczne w B2B
`deposit_pack_multiplier`, `deposit_pack_price`, `deposit_pack_type`	opakowanie zwrotne (kaucja): mnożnik, cena, rodzaj

Kategoria

```
{ "id": 12, "name": "Zapachy samochodowe", "description": "", "category_external_id": "KAT-012" }
```

`category_external_id` to ID kategorii w systemie ERP.

Przypisanie produktu do kategorii

```
{ "product_id": 1037, "category_id": 12 }
```

Zdj?cie produktu

```
{ "product_id": 1037, "image": "https://.../products/850267.jpg", "order": 1, "product_external_id": "850267" }
```

`order` to kolejność zdjęć – zdjęcie z najniższą wartością jest główne.

Stan magazynowy

Jeden wiersz na produkt, zsumowany ze wszystkich magazynów:

```
{ "product_id": 1037, "availability": "medium", "quantity": 150 }
```

Pole	Opis
`availability`	stan słownie: `none` (brak), `low` (poniżej 100), `medium` (100–999), `high` (1000 i więcej)

Pole	Opis
`quantity`	rzeczywista ilość, gdy klient ma włączoną opcję *Zwracaj rzeczywistą wartość stanów w API i ofertach*; w przeciwnym razie `1` (jest na stanie) lub `0`

Rodzaj wysy?ki

```
{ "id": 3, "name": "Kurier" }
```

`id` przekazujesz jako `shipment_type_id` przy tworzeniu zamówienia.

Kody odpowiedzi

Kod	Znaczenie
`200`	poprawna odpowiedź
`401`	brak klucza albo klucz wygasł lub został wyłączony
`404`	nie ma obiektu o podanym `id` (albo nie jest dostępny dla klienta)
`429`	przekroczony limit jednoczesnych zapytań lub zapytań na minutę – ponów po czasie z nagłówka `Retry-After`

Przyk?ady

W przykładach zastąp `{amper_ws_url}` i `{amper_api_key}` swoimi danymi.

curl

```
curl "https://{amper_ws_url}/external/v1/products/?api_key={amper_api_key}&limit=500&offset=0"
```

Python – pobranie całego katalogu stronami, z obsługą limitu:

```
import time


import requests


BASE_URL = "https://{amper_ws_url}/external/v1"
params = {"api_key": "{amper_api_key}", "limit": 500, "offset": 0}
```

```
products = []
while True:
    response = requests.get(f"{BASE_URL}/products/", params=params, timeout=120)
    if response.status_code == 429:
        time.sleep(int(response.headers.get("Retry-After", 1)))
        continue
    response.raise_for_status()
    page = response.json()
    products.extend(page["results"])
    if not page["next"]:
        break
    params["offset"] += params["limit"]

print(len(products))
```

C# (.NET, `HttpClient`)

```
using var client = new HttpClient { Timeout = TimeSpan.FromMinutes(2) };
var url =
"https://{amper_ws_url}/external/v1/products/?api_key={amper_api_key}&limit=500&offset=0";
var json = await client.GetStringAsync(url);
Console.WriteLine(json);
```

Node.js (18 i nowsze, wbudowany `fetch`)

```
const url =
'https://{amper_ws_url}/external/v1/products/?api_key={amper_api_key}&limit=500&offset=0';
const response = await fetch(url);
if (!response.ok) throw new Error(`HTTP ${response.status}`);
const page = await response.json();
console.log(page.count, page.results.length);
```

PHP (cURL)

```
$curl =
curl_init('https://{amper_ws_url}/external/v1/products/?api_key={amper_api_key}&limit=500&offs
et=0');
curl_setopt_array($curl, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_TIMEOUT => 120,
]);
```

```
$response = curl_exec($curl);
curl_close($curl);
$page = json_decode($response, true);
echo $page['count'];
```