Promocje - integracja z systemami zewn?trznymi

? Dokumentacja techniczna struktura i logika promocji

? 1. Struktura pojedynczej promocji

Każdy element tablicy to *jedna promocja* w systemie AMPER B2B.


```
{
  "id": 22960679,
  "name": "promocja na palete",
  "description": "",
  "start": "2025-10-23T00:00:00+02:00",
  "end": "2028-10-20T23:59:00+02:00",
  "short_code": "promocja-na-palete",
  "default_image": { "image": "...", "alt": "..." },
  "keep_bundle_multiplayer_same": false,
  "use_promotion_price_for_threshold": false,
  "calculate_using_cumulative_units": true,
  "multiply_reward": 0,
  "color_in_cart": null,
  "start_using_default_price": false,
  "is_active": true,
  "conditions": [ ... ]
}
```


? Kluczowe pola

Pole	Typ	Opis
id	int	ID promocji
name	string	Nazwa promocji
description	string	Opis (opcjonalny)
start, end	datetime	Zakres obowiązywania promocji
short_code	string	Jednoznaczny kod identyfikacyjny

Pole	Typ	Opis
default_image	object	Domyślna grafika promocji
keep_bundle_multiplayer_same	bool	Czy utrzymać wielokrotność w zestawach
use_promotion_price_for_threshold	bool	Czy threshold/próg liczony wg ceny promocyjnej
calculate_using_cumulative_units	bool	Czy przeliczać jednostki po konwerterach/jednostkach zbiorczych
multiply_reward	int	Czy nagroda mnożona przez ilość pakietów
color_in_cart	string/null	Kolor w koszyku
start_using_default_price	bool	Czy liczyć od ceny domyślnej
is_active	bool	Czy aktywna
conditions	array	Lista warunków aktywujących promocję

? 2. Warunki promocji (conditions)

Każda promocja może zawierać **jeden lub wiele warunków**.

Struktura warunku:

```
{
  "threshold_min": 1.0,
  "threshold_max": 0.0,
  "measure": "quantity",
  "rewards": [ ... ]
}
```

? Pola warunku

Pole	Typ	Opis
threshold_min	float	Minimalny próg osiągnięcia promocji
threshold_max	float	Maksymalny próg (0 = brak limitu)
measure	string	Typ pomiaru: "quantity" lub "value"
rewards	array/null	Lista nagród (ceny, gratisy, rabaty)

? 3. Nagrody (rewards)

Każda nagroda jest powiązana do konkretnego produktu.

```
{
  "given_price": 8.0,
  "reward_value": 0.0,
  "value_type": "3",
  "quantity": 0.0,
  "product": {
    "id": 817,
    "ean": "817",
    "name": "BATERIA ENERGIZER...",
    "short_code": "426684-CC",
    "default_unit_of_measure": "bl.",
    "cumulative_unit_of_measure": "karton",
    "cumulative_converter": "24.00",
    "can_be_split": true,
    "cumulative_unit_ratio_splitter": "24.00",
    "unit_roundup": false
  }
}
```


? Pola nagrody

Pole	Typ	Opis
given_price	float	Cena promocyjna produktu
reward_value	float	Wartość rabatu (jeśli dotyczy)
value_type	string	Typ rabatu (np. stała cena, gratis, procent)
quantity	float	Ilość nagrody (np. liczba gratisów)
product	object	Opis produktu objętego promocją

? 4. Struktura produktu w nagrodzie (product)

Pole	Typ	Opis
id	int	ID produktu
ean	string	Kod EAN
name	string	Pełna nazwa
short_code	string	Kod skrócony
default_unit_of_measure	string	Jednostka bazowa
cumulative_unit_of_measure	string	Jednostka zbiorcza

Pole	Typ	Opis
cumulative_converter	string	Przelicznik jednostek
can_be_split	bool	Czy można dzielić jednostki
cumulative_unit_ratio_splitter	string	Współczynnik dzielenia
unit_roundup	bool	Czy zaokrąglać w górę

? 5. Logika liczenia promocji

5.1. Typ progu – measure

Dwa tryby:

1) quantity

Liczymy ilość sztuk / opakowań / jednostek.

2) value

Liczymy wartość koszyka dla produktów spełniających warunki.

5.2. Przeliczanie jednostek (calculate_using_cumulative_units)

Jeśli **true** – ilość do thresholdu liczona jest tak:

```
ilość_w_sztukach * cumulative_converter
```

Przykład:

- produkt: 6-pack → cumulative_converter = 6
- klient kupuje 2 opakowania
- do thresholdu liczymy: **12 szt.**

5.3. Multiplikacja nagród (multiply_reward)

Jeśli multiply_reward > 0:

- osiągnięcie kolejnych progów zwiększa liczbę nagród

Przykład:

threshold = 10 szt

multiply_reward = 1

klient kupił 30 szt → **3x nagroda**

? 6. Typowe przypadki

? Sta?a cena za produkt

Promocje typu:

> reward → value_type = 3, given_price > 0

Przykład: “Stała cena 5 zł przy zakupie min 10 szt”

? Gratis

Promocja ma quantity > 0 oraz given_price = 0.

? Progi wielokrotno?ci

Promocje:

- Próg wielokrotności - produkt kwota
- Próg wielokrotności - kategoria kwota

Mechanika:

- measure = value
- wiele warunków z tym samym threshold
- wiele nagród równoległych

? 7. Przyk?ad – schematy strukturalne

Promocja (Promo)

```
Promo
 ├── conditions [Condition]
 │     ├── threshold_min
 │     ├── threshold_max
 │     ├── measure (quantity/value)
 │     └── rewards [Reward]
 │           ├── given_price
 │           ├── reward_value
 │           ├── value_type
 │           ├── quantity
 │           └── product (Product)
 └── metadata...
```

? 8. U?ycie

Algorytm wyliczania promocji:

1. Pobierz wszystkie aktywne promocje dla klienta
2. Dla każdej promocji:
   - sprawdź daty start/end
   - oblicz threshold dla danego measure
   - uwzględnij cumulative conversions
3. Jeśli warunki spełnione:
   - zastosuj nagrody:
     - nadpisz cenę (given_price)
     - dodaj gratisy
     - nadaj rabat procentowy/kwotowy
4. Zapisz informację o zastosowanej promocji w koszyku.

? 9. U?ycie po stronie frontend (koszyk)

Front powinien:

- wyróżnić produkty w promocji (color_in_cart)
- pokazać cenę promocyjną given_price
- pokazać opis promocji
- dodać gratisy jako osobne linie (jeśli quantity > 0)
- opcjonalnie pokazać osiągnięty próg

? 10. Podsumowanie

AMPER to elastyczny model promocji zawierający:

✔ wiele progów

✔ wiele typów nagród

✔ wielokrotność progów

✔ konwersje jednostek

✔ stałe ceny

✔ gratisy

✔ rabaty

✔ pakiety

Revision #3
Created 28 November 2025 12:21:38 by Tomek Dziemidowicz
Updated 28 November 2025 12:36:20 by Tomek Dziemidowicz