

Tworzenie zamówienia "create-order"

Przebieg

Endpoint `POST /create-order` tworzy zamówienie na podstawie podanych danych klienta i linii produktów. Przed wywołaniem należy uzyskać token JWT przez endpoint autoryzacyjny.

1. Autoryzacja - uzyskanie tokena

1.1 Pobranie tokena (logowanie)

```
POST /auth/  
Content-Type: application/json
```

Body żądania:

```
{  
  "username": "login_uzytkownika",  
  "password": "haslo"  
}
```

Przykład (curl):

```
curl -X POST https://<host>/auth/ \  
  -H "Content-Type: application/json" \  
  -d '{"username": "jan.kowalski", "password": "tajnehaslo"}'
```

Odpowiedź (200 OK):

```
{  
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ikpz...",  
  "expires_in": 300,
```

```
"refresh_expires_in": 1800,  
"refresh_token": "eyJhbGciOiJIUzI1NiIsInR5c...",  
"token_type": "Bearer",  
"not-before-policy": 0,  
"session_state": "abc123...",  
"scope": "amper"  
}
```

Pole	Opis
access_token	Token JWT do autoryzacji żądań API
expires_in	Czas ważności tokena w sekundach (domyślnie 300 s = 5 min)
refresh_token	Token do odświeżenia sesji bez ponownego logowania
refresh_expires_in	Czas ważności refresh tokena w sekundach

1.2 Odświeżenie tokena

Gdy `access_token` wygaśnie, użyj `refresh_token` zamiast ponownie logować użytkownika.

```
POST /auth/token-refresh/  
Content-Type: application/json
```

Body żądania:

```
{  
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5c..."  
}
```

Przykład (curl):

```
curl -X POST https://<host>/auth/token-refresh/ \  
-H "Content-Type: application/json" \  
-d '{"refresh_token": "eyJhbGciOiJIUzI1NiIsInR5c..."}'
```

Odpowiedź: Taki sam format jak przy logowaniu — nowy `access_token` i `refresh_token`.

2. Tworzenie zamówienia

Endpoint

POST /create-order

Authorization: Bearer <access_token>

Content-Type: application/json

Wymagane uprawnienia

Token musi należeć do użytkownika z zakresem `B2B` lub `TRANSLATOR`.

2.1 Parametry zapytania

Body zapytania (JSON):

Pole	Typ	Wymagane	Opis
<code>customer_id</code>	<code>integer</code>	Tak*	ID klienta w systemie
<code>customer_external_id</code>	<code>string</code>	Tak*	Zewnętrzny ID klienta (alternatywa dla <code>customer_id</code>)
<code>lines</code>	<code>array</code>	Tak	Lista linii zamówienia (patrz niżej)
<code>playground</code>	<code>boolean</code>	Nie	<code>true</code> — symulacja bez zapisu zamówienia (domyślnie <code>false</code>)
<code>description</code>	<code>string</code>	Nie	Opis/notatka do zamówienia
<code>shipment_type_id</code>	<code>integer</code>	Nie	ID typu wysyłki

“* Wymagane jest podanie `customer_id` lub `customer_external_id`.

Pola obiektu w tablicy `lines`:

Pole	Typ	Wymagane	Opis
<code>product_id</code>	<code>integer</code>	Tak*	ID produktu w systemie
<code>product_external_id</code>	<code>string</code>	Tak*	Zewnętrzny ID produktu (alternatywa dla <code>product_id</code>)

Pole	Typ	Wymagane	Opis
quantity	decimal	Tak	Ilość (zaokrąglana do kroku jednostki produktu)
suggested_price	decimal	Tak	Sugerowana cena jednostkowa
promotion_id	integer	Nie	ID konkretnej promocji do zastosowania
order_line_promotion_id	integer	Nie	ID promocji linii zamówienia

“* Wymagane jest podanie `product_id` **lub** `product_external_id`.

2.2 Przyk?ady ??da?

Przyk?ad z wewn?rznymi ID:

```
curl -X POST https://<host>/create-order \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6Ii..." \
-H "Content-Type: application/json" \
-d '{
  "customer_id": 42,
  "lines": [
    {
      "product_id": 101,
      "quantity": 10,
      "suggested_price": 99.99
    },
    {
      "product_id": 205,
      "quantity": 5,
      "suggested_price": 49.50,
      "promotion_id": 7
    }
  ],
  "description": "Zamówienie testowe"
}'
```

Przyk?ad z zewn?rznymi ID (integracja ERP):

```
curl -X POST https://<host>/create-order \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cC..." \  
-H "Content-Type: application/json" \  
-d '{  
  "customer_external_id": "ERP-KLIENT-0042",  
  "lines": [  
    {  
      "product_external_id": "SKU-ABC-001",  
      "quantity": 10,  
      "suggested_price": 99.99  
    }  
  ]  
'
```

Przykład trybu playground (symulacja bez zapisu):

```
curl -X POST https://<host>/create-order \  
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cC..." \  
-H "Content-Type: application/json" \  
-d '{  
  "customer_id": 42,  
  "playground": true,  
  "lines": [  
    {  
      "product_id": 101,  
      "quantity": 10,  
      "suggested_price": 99.99  
    }  
  ]  
'
```

2.3 Odpowiedź?

Sukces — 201 Created:

```
{  
  "order_id": 12345,  
  "lines": [  
    {
```

```

"product_id": 101,
"product_external_id": "SKU-ABC-001",
"quantity": 10,
"suggested_price": 99.99,
"default_price": 100.00,
"customer_price": 95.00,
"given_price": 90.00,
"promotion": {
  "price": 90.00,
  "promotion_min_order_quantity": 5,
  "promotion_condition_item_id": 123,
  "promotion_condition_id": 456,
  "relation_id": 789,
  "promotion_name": "Letnia wyprzedaż",
  "promotion_id": 999,
  "promotion_url": "https://<host>/promotions/999/"
}
}
]
}

```

Pole	Opis
<code>order_id</code>	ID utworzonego zamówienia; <code>0</code> gdy <code>playground=true</code> lub brak linii
<code>lines</code>	Linie zamówienia z przeliczonymi cenami i promocjami
<code>lines[].default_price</code>	Cena katalogowa
<code>lines[].customer_price</code>	Cena indywidualna klienta
<code>lines[].given_price</code>	Cena końcowa (po uwzględnieniu promocji; <code>0</code> gdy brak promocji)
<code>lines[].promotion</code>	Najlepsza dostępna promocja lub <code>null</code> gdy nie dotyczy

“ **Uwaga:** W odpowiedzi zwracane są tylko linie produktów, które są dostępne dla danego klienta i mają prawidłowe dane cenowe. Linie z niedostępnymi produktami są pomijane.

3. Kody b??dów

Kod	Opis
201 Created	Zamówienie zostało pomyślnie utworzone
401 Unauthorized	Brak lub nieprawidłowy token <code>Authorization</code>
403 Forbidden	Token ważny, ale brak wymaganego zakresu (<code>B2B</code> lub <code>TRANSLATOR</code>)
404 Not Found	Nieprawidłowe dane logowania przy pobieraniu tokena

4. Pełny przepływ integracji

1. POST `/auth/` → otrzymujesz `access_token` + `refresh_token`
|
▼
2. POST `/create-order` → zamówienie zostaje utworzone
`Authorization: Bearer <access_token>`
|
▼
3. Gdy `access_token` wygaśnie (`expires_in` sekund):
POST `/auth/token-refresh/` → otrzymujesz nowy `access_token`
{ "refresh_token": "..." }

Rekomendowane podejście w kliencie:

- Przechowuj `access_token` i `refresh_token` w pamięci sesji.
- Przed każdym żądaniem sprawdzaj, czy token nie wygał (`expires_in`).
- W przypadku odpowiedzi `401` automatycznie wywołaj endpoint odświeżenia tokena.
- Gdy `refresh_token` również wygaśnie, ponownie pobierz token przez `/auth/`.

5. Uwagi implementacyjne

- **Tryb playground** (`"playground": true`) pozwala przetestować logikę cenową i promocyjną bez tworzenia zamówienia w bazie danych. Zwrócone `order_id` będzie równe `0`.
- **Zewnętrzne ID** (`customer_external_id`, `product_external_id`) są tłumaczone na wewnętrzne ID systemu przez moduł translatora. Używaj ich przy integracji z zewnętrznymi systemami (ERP, WMS).
- **Ilość** (`quantity`) jest automatycznie zaokrąglana do kroku jednostki produktu (np. jeśli krok to 5, zamówienie 7 szt. zostanie zaokrąglone do 10).

- **Promocje** są wybierane automatycznie (najlepsza dostępna dla klienta), chyba że podasz `promotion_id` w linii zamówienia.
-

Revision #5

Created 28 November 2025 12:20:29 by Tomek Dziemidowicz

Updated 10 June 2026 08:22:19 by Tomek Dziemidowicz