



PayPo API v.3.0

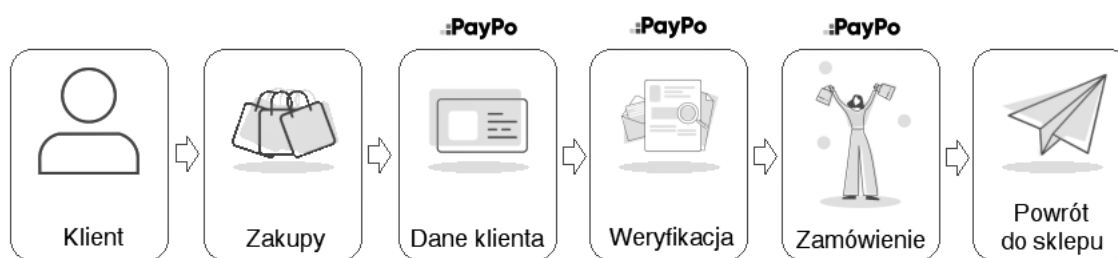
Dokument zawiera specyfikację techniczną REST API PayPo.pl w wersji 3.0

Wersja dokumentu	Data	Wykaz zmian
3.0.7.	2022.04.26	Poprawki w zakresie nowych pól dla adresów.
3.0.6.	2021.11.08	Poprawki w zakresie nowych pól dla konfiguracji produktu.
3.0.5	2021.08.03	Poprawki w zakresie powiadomień
3.0.4	2020.10.02	Poprawki definicji i endpointów
3.0_r3	2020.07.16	Powiadomienia o zmianie statusu i kwoty zamówienia
3.0_r2	2020.06.01	Uwierzytelnienie komunikatów OAuth
3.0_r1	2020.05.12	Doprecyzowanie opisu atrybutów adresowych, autoryzacja HMAC
3.0_r0	2020.04.28	Definicje podstawowych endpointów, cykl życia transakcji.

1. Opis usługi

PayPo świadczy usługę odroczonej płatności dla klientów sklepów internetowych (Kupujących). Rejestracja nowej transakcji jest realizowana online z wykorzystaniem API, analogicznie jak w popularnych systemach płatności.

Przebieg transakcji



- Kupujący po wybraniu produktów wybiera odroczoną płatność za zakupy z PayPo.
- Dane Kupującego są przekazywane do serwisu PayPo.
- System PayPo weryfikuje wiarygodność finansową Kupującego.
- Pozytywna weryfikacja kończy się odroczeniem płatności za zamówienie.
- Kupujący jest przekierowany zwrótnie na stronę sklepu.

Dalsze kroki

- Sprzedający potwierdza realizację zamówienia w systemie PayPo.
- Przelew środków do Sprzedającego następuje po aktualizacji statusu zamówienia w terminie określonym w umowie.
- Kupujący opłaca koszty zakupów do PayPo zgodnie z warunkami odroczenia zapłaty PayPo.

2. Integracja – informacje ogólne

API dostępne jest pod adresem: <https://api.paypo.pl/v3/>

Adres serwera testowego: <https://api.sandbox.paypo.pl/v3/>

Domena zawarta w linkach API jest uzależniona od kraju, np. dla Rumunii link będzie skonstruowany - <https://api.paypo.ro/v3/>

PayPo API bazuje na architekturze REST. Komunikacja realizowana jest z wykorzystaniem protokołu HTTP (metody POST/GET/PATCH), dane wymieniane są w formacie JSON. Podstawowym zasobem udostępnionym Sprzedającym są transakcje (/transactions).

Przykładowa rejestracja płatności:

```
curl -X POST https://api.paypo.pl/v3/transactions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer fLEsy5wk9jQKK0NjbF57sbOwa4BdCPul" \
-d '{
  "id": "123abc",
  "shopId": "012abc",
  "order": {
    "referenceId": "ord_98765/20",
    "providerId": "180abc",
    "description": "test",
    "additionalInfo": ["additionalInfo",]
    "amount": 24900,
    "billingAddress": {
      "street": "Kredytowa",
      "building": "9a",
      "flat": "3",
      "zip": „00-950",
      "city": "Warszawa"
      "county": "mazowieckie"
      "country": "PL"
    },
    "shippingAddress": {
      "street": "Domaniewska",
      "building": "37",
      "flat": "",
      "zip": "02-672",
      "city": "Warszawa"
      "county": "mazowieckie"
      "country": "PL"
    },
    "shipment": 0,
  },
  "customer": {
    "name": "Anna",
    "surname": "Nowak",
    "email": "anna.n@paypo.pl",
    "phone": "+48500123456"
  },
  "configuration": {
    "returnUrl": "https://merchant.com/complete",
    "notifyUrl": "https://merchant.com/notify",
    "cancelUrl": https://merchant.com/cancel
  }
}
```

```

    "product": {
      "productType": "core",
      "process": "online",
      "installmentCount": 0
    }
  }
}'

```

3. Tworzenie nowego zamówienia

3.1 Rejestracja płatności

Adres URL: <https://api.paypo.pl/v3/transactions>

W celu utworzenia nowej sesji płatności należy przekazać do PayPo podstawowe dane transakcji. Po prawidłowej weryfikacji komunikatu w odpowiedzi przekazany zostanie adres URL formularza weryfikacyjnego, pod który należy przekierować Kupującego.

Po weryfikacji tożsamości i potwierdzeniu wiarygodności finansowej PayPo kieruje Kupującego zwrótnie na stronę sklepu (`returnUrl`) oraz wysyła powiadomienie o zmianie statusu zamówienia. Cykl życia transakcji został opisany w p. 4. dokumentacji.

Parametry wejściowe żądania (HTTP POST):

Nazwa parametru	Typ	Wymagany	Opis	Sposób walidacji
id	uuid	N	Identyfikator transakcji	-
shopId	uuid	N	Identyfikator sklepu nadany przez PayPo (opcjonalnie w przypadku kiedy Sprzedawca lub Operator Płatności integruje osobne punkty sprzedaży)	Czy wartość w formacie UUID
order	{}	T	Dane zamówienia	-
referencId	string	T	Identyfikator transakcji po stronie sprzedawcy, wykorzystywany do identyfikacji przelewów	Czy pole nie jest puste
providerId	string	N	Identyfikator transakcji w systemie Operatora Płatności (przy integracji przez Operatora)	-
description	string	N	Opis / tytuł transakcji	-

	additionalInfo		{}	N	Dodatkowy opis zamówienia, np. zawartość koszyka	-
	amount		int	T	Kwota zamówienia w podstawowych jednostkach monetarnych, np. groszach	Czy wartość większa od '0'
	shipment		int	N	Metoda dostawy: 0 - kurier (domyślnie) 1 - dostawa do punktu (np. UPS Access point, DHL Parcel Shop) 2 - paczkomat 3 - paczka w RUCHu 4 - odbiór w sklepie (click&collect)	Czy wartość zgodna z możliwymi wartościami pochodzącymi z przedziału <0,4>
	billingAddress			T	Adres rozliczeniowy:	-
		street	string	T	Ulica	Czy wartość nie jest pusta
		building	string	N	Numer domu	-
		flat	string	N	Numer lokalu	-
		zip	string	T	Adres rozliczeniowy: kod pocztowy	Czy wartość pasuje do wzorca regex /^\d+--\d+\$/
		city	string	T	Adres rozliczeniowy: miasto	Czy wartość nie jest pusta
		county	string	N	Adres rozliczeniowy: województwo	-
		country	string	N	Adres rozliczeniowy: Kod kraju, np. PL, RO (domyślnie PL)	Czy wartość zgodna z ISO 3166-1 alpha-2
	shippingAddress			T	Adres dostawy:	-
		street	string	T	Ulica	Czy wartość nie jest pusta
		building	string	N	Numer domu	-

		flat	string	N	Numer lokalu	-
		zip	string	T	Kod pocztowy	Czy wartość pasuje do wzorca regex /^\d+-\d+\$/
		city	string	T	Miasto	Czy wartość nie jest pusta
		county	string	N	Województwo	-
		country	string	N	Kod kraju, np. PL, RO (domyślnie PL)	Czy wartość zgodna z ISO 3166-1 alpha-2
customer			{}	T	Dane Kupującego	-
	name		string	T	Imię	Czy wartość nie jest pusta
	surname		string	T	Nazwisko	Czy wartość nie jest pusta
	email		string	T	Adres email	Czy wartość jest poprawna
	phone		string	N	Numer telefonu komórkowego z uwzględnieniem prefixu kraju. W braku wskazania prefixu domyślnie zostanie przyjęty dla kraju wskazanego w polu 'locale'	-
configuration			{}	T	Adresy powrotu i notyfikacji	-
	returnUrl		string	T	Adres powrotu po przetworzeniu płatności	Poprawny URL
	notifyUrl		string	T	Adres, na który wysyłane będą powiadomienia	Poprawny URL
	cancelUrl		string	N	Opcjonalny adres powrotu w przypadku rezygnacji Kupującego z płatności PayPo, jeżeli nie zostanie określony, to domyślnie użyty zostanie returnUrl	Jeżeli wartość wskazana to poprawny URL
	product				Konfiguracja produktu	-

		productType	string	N	Typ produktu: <ul style="list-style-type: none"> • core • PNX 	-
		process	string	N	Typ procesu	-
		installmentCount	int	N	Ilość rat	-

Uwierzytelnienie żądań zostało opisane szczegółowo w p. 9. dokumentacji. W wersji API 3.0.x wymagane jest przekazywanie w nagłówku komunikatu tokenu uwierzytelniającego OAuth.

3.2 Odpowiedź serwera

W odpowiedzi na wywołanie metody /transactions PayPo zwraca status operacji oraz URL formularza, pod który należy przekierować Kupującego.

Przykładowa odpowiedź serwera:

```
HTTP 201 Created
{
  "transactionId": "5909da74-af95-41e9-b8e2-12e61c3c6f27",
  "redirectUrl": "https://process.paypo.pl/5909da74-af95-41e9-b8e2-12e61c3c6f27"
}
```

3.3 Weryfikacja danych Kupującego i powrót do sklepu

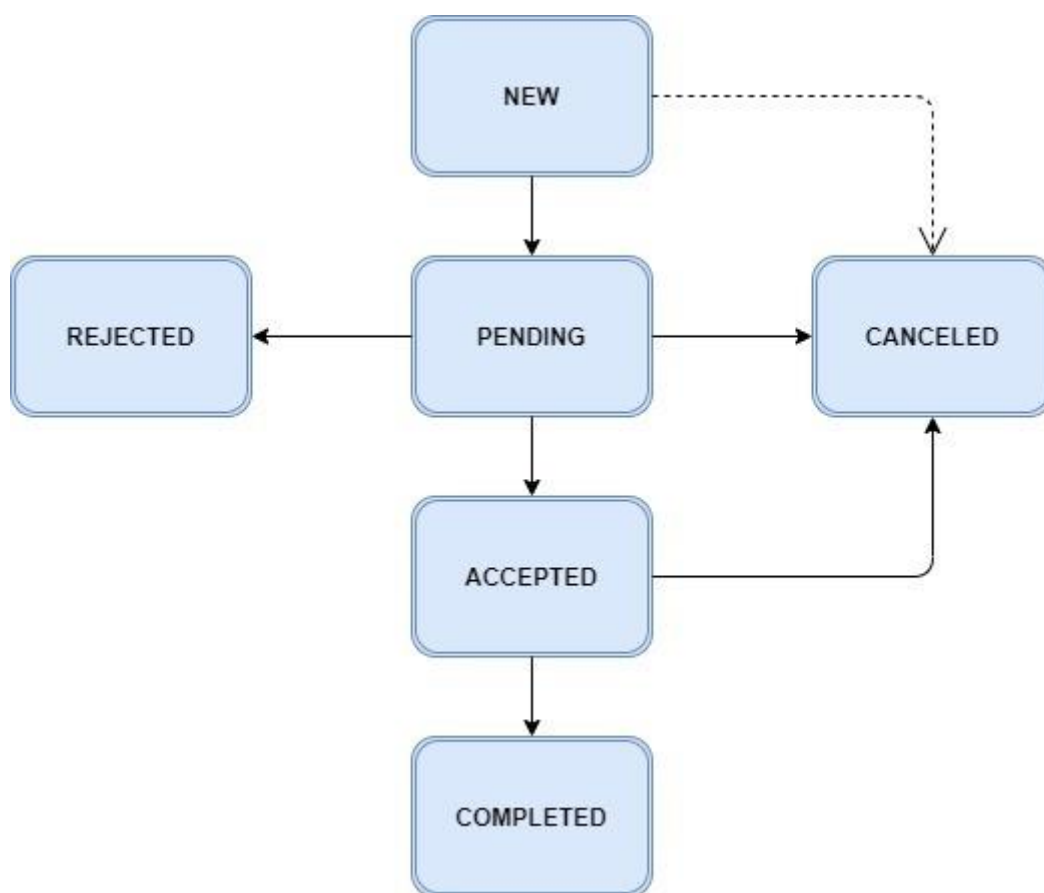
Po przekierowaniu do PayPo następuje uzupełnienie i weryfikacja danych Kupującego. Po przeprosowaniu płatności Kupujący jest przekierowywany zwrótnie na adres returnUrl. W parametrach GET adresu powrotnego przekazywany jest status płatności, w celu wyświetlenia odpowiedniego komunikatu w przeglądarce.

[https://merchant.com/thankyou?status=\[OK|ERR\]](https://merchant.com/thankyou?status=[OK|ERR])

Jeżeli w komunikacie inicjującym transakcję został dodatkowo przesłany atrybut returnUrl, rezygnacja w trakcie procesu weryfikacji skutkuje przekierowaniem na ten adres (np. ekran wyboru alternatywnej metody płatności przy checkoucie).

4. Statusy i cykl życia transakcji

Statusy płatności:



Status	Opis
NEW	Nowa transakcja
PENDING	Płatność w trakcie procesowania, oczekuje na potwierdzenie tożsamości Kupującego lub potwierdzenie dostępności środków
ACCEPTED	Płatność zaakceptowana
COMPLETED	Transakcja potwierdzona przez Sprzedawcę, płatność gotowa do rozliczenia
REJECTED	Odmowa realizacji płatności
CANCELED	Transakcja anulowana

Po prawidłowej rejestracji płatność przyjmuje status NEW, po przekierowaniu Kupującego ustawiany jest status PENDING, następnie transakcja może zostać zaakceptowana (ACCEPTED) lub odrzucona (REJECTED) przez system PayPal.

Transakcja może także zostać zaakceptowana (ACCEPTED) po wcześniejszym jej odrzuceniu (REJECTED). Możliwe jest to w sytuacji, gdy Kupującemu zostanie udostępniony link umożliwiający powrót do płatności i PayPal udzieli odroczonej płatności.

Anulowanie sesji płatności dozwolone jest przed jego potwierdzeniem, przed przejściem w status COMPLETED.

Podczas weryfikacji statusów należy uwzględnić ich hierarchię, np. w przypadku otrzymania statusu COMPLETED należy uznać, że statusy NEW, PENDING i ACCEPTED zostały już przesłane. Analogicznie sytuacja wygląda dla statusów REJECTED i CANCELED – statusy NEW i PENDING uznać należy za otrzymane.

5. Powiadomienia

5.1 Zmiana statusu płatności lub kwoty transakcji

Po zmianie statusu płatności lub kwoty transakcji szczegółowa informacja wysyłana jest w potwierdzeniu na adres `notifyUrl`. System Sprzedającego w odpowiedzi na powiadomienie powinien zwrócić HTTP status 200.

Parametry notyfikacji (HTTP POST):

Nazwa parametru	Typ	Opis
merchantId	uuid	Identyfikator sprzedawcy
shopId	uuid	Identyfikator POS (jeśli był zdefiniowany przy rejestracji transakcji)
referenceId	string	Identyfikator transakcji po stronie sprzedawcy
transactionId	uuid	Identyfikator transakcji PayPo
transactionStatus	string	Status zamówienia, zgodnie z tabelą w p. 4, wyjątkiem jest status 'NEW', który nie jest przekazywany
transactionUrl	string	Link z adresem do płatności odroczonej
amount	int	Aktualna kwota transakcji podstawowych jednostkach monetarnych, np. groszach
lastUpdate	timestamp	Data aktualizacji w formacie RFC3339

Notyfikacje zawierają w nagłówkach dodatkowo typ dokumentu oraz skrót uwierzytelniający. Algorytm wyliczania i weryfikacji skrótu został opisany w p.9.2. dokumentacji.

Przykładowa notyfikacja:

```

POST /notifyUrl HTTP/1.1
X-PayPo-Signature: dfcffcd0cbcd7bbbd11b7d23f3a2e7a7458ecf90
{
  "merchantId": "19c692be-a893-468c-a65f-b8de442e5443",
  "referenceId": "ord_987654",
  "transactionId": "00102030",
  "transactionStatus": "PENDING",
  "amount": 24900,
  c: „2020-03-05T10:54:02+01:00“,
}

```

W przypadku braku odpowiedzi lub odebrania statusu innego niż HTTP 20x system PayPo podejmie próbę ponowienia notyfikacji zgodnie z poniższym schematem (konfiguracja ponowień może wyglądać inaczej w środowiskach nieprodukcyjnych).

Pierwsza godzina po poprawnie zweryfikowanej transakcji	Co 10 minut
Kolejne 5h	Co 20 minut
Kolejne 18h	Co 60 minut

5.2 Wygenerowanie specyfikacji do przelewu

W chwili wygenerowania specyfikacji do przelewu wysyłana jest notyfikacja na adres `notifyUrl`. System w odpowiedzi na powiadomienie powinien zwrócić HTTP status 200 w przypadku pozytywnego otrzymania notyfikacji oraz odpowiedni status 4xx lub 5xx w przypadku wystąpienia błędu.

Funkcjonalność udostępniania jest na prośbę Sprzedającego. Domyślnie jest wyłączona.

Parametry notyfikacji (HTTP POST):

Nazwa parametru	Typ	Wymagany	Opis
merchantId	uuid	T	Identyfikator sprzedawcy
referenceId	string	T	Identyfikator transakcji po stronie sprzedawcy
transactionId	uuid	T	Identyfikator transakcji PayPo
transactionStatus	string	T	Status zamówienia, zgodnie z tabelą w p. 4
amount	int	T	Aktualna kwota transakcji podstawowych jednostkach monetarnych, np. groszach

lastUpdate	timestamp	T	Data aktualizacji w formacie RFC3339
settlementStatus	string	T	Status płatności, zgodnie z tabelą w p. 7
message	string	T	Dodatkowa informacja
shopId	uuid	T	Identyfikator POS (jeśli był zdefiniowany przy rejestracji transakcji)

Przykład notyfikacji:

```
{
  "merchantId": "0e1576d8-e760-4336-8bc4-c20a549ac035",
  "referenceId": "QQBF6HAWVGI972291WQQ",
  "transactionId": "9207c39a-1d1f-4954-a312-fe5dcd1a1f8a",
  "transactionStatus": "COMPLETED",
  "amount": 1212,
  "lastUpdate": "2021-07-27T18:44:51.000+02:00",
  "settlementStatus": "PAID",
  "message": "Transaction is settled",
  "shopId": "c8847732-b5d7-4528-8a7b-54ede5e43789"
}
```

6. Potwierdzenie transakcji

Po pozytywnej weryfikacji płatności, przesłaniu notyfikacji i przekierowaniu Kupującego na stronę sklepu, system PayPal oczekuje na potwierdzenie realizacji zamówienia przez Sprzedawcę. W celu potwierdzenia zamówienia system Sprzedawcy powinien przesłać żądanie zmiany statusu na COMPLETED. Istnieje możliwość automatycznego potwierdzania każdego zamówienia z zaakceptowaną płatnością (parametr autodelivery), konfiguracja jest przygotowywana indywidualnie na życzenie Sprzedawcy. Dla uruchomionej usługi bramki płatniczej PayPal parametr autodelivery jest wymagany.

Niepotwierdzone zamówienia mogą zostać automatycznie anulowane przez PayPal po określonym czasie, konfiguracja przygotowywana jest na życzenie Sprzedawcy.

Adres URL: [https://api.paypo.pl/v3/transactions/\\${UUID}](https://api.paypo.pl/v3/transactions/${UUID})

Parametry wejściowe (HTTP PATCH):

Nazwa parametru	Wartość	Opis
status	COMPLETED	Potwierdzenie realizacji zamówienia

Prawidłowe przyjęcie żądania zostanie potwierdzone odpowiedzią ze statusem 200. Po aktualizacji statusu płatności odpowiednia notyfikacja zostanie przesłana na adres `notifyUrl` przekazany przy rejestracji płatności.

Przykładowa odpowiedź:

```
HTTP 200 OK
{
  "code": 200,
  "message": "Transaction updated successfully",
}
```

7. Weryfikacja statusu płatności

Aktualny status zamówienia można odczytać w dowolnym momencie wywołując metodę HTTP GET. Żądanie nie wymaga żadnych parametrów poza przekazanym w URLu identyfikatorem płatności. Zapytanie powinno być uwierzytelnione tokenem OAuth.

Adres URL: [https://api.paypo.pl/v3/transactions/\\${UUID}](https://api.paypo.pl/v3/transactions/${UUID})

Przykładowa odpowiedź:

```
HTTP 200 OK
{
  "merchantId": "19c692be-a893-468c-a65f-b8de442e5443",
  "referenceId": "ord_987654",
  "transactionId": "cd975bc6-a755-4141-b7a0-d7e8f7a308ef",
  "transactionStatus": "COMPLETED",
  "amount": 24900,
  "settlementStatus": "PAID",
  "lastUpdate": "2020-03-05T10:54:02",
}
```

W odpowiedzi oprócz podstawowych atrybutów transakcji przekazywany jest również status rozliczenia płatności `settlementStatus`, może przyjmować jedną z trzech wartości:

Status rozliczenia	Opis
NEW	Nowa transakcja
CONFIRMED	Potwierdzono realizację zamówienia, transakcja gotowa do rozliczenia
PAID	Transakcja rozliczona za Sprzedawcą

8. Modyfikacja atrybutów transakcji

Sprzedający może modyfikować status transakcji oraz kwotę zamówienia. Dozwolone zmiany statusu to anulowanie (CANCELED) oraz potwierdzenie (COMPLETED, opisane wyżej). Do modyfikacji kwoty zamówienia służy dedykowana metoda rejestrowania zwrotów /refunds.

8.1 Anulowanie płatności i draftów

Adres URL: [https://api.paypo.pl/v3/transactions/\\${UUID}](https://api.paypo.pl/v3/transactions/${UUID})

Parametry wejściowe (HTTP PATCH):

Nazwa parametru	Wartość	Opis
status	CANCELED	Anulowanie zamówienia

Anulowanie płatności może zostać zlecone przez Sprzedającego w przypadku braku możliwości realizacji zamówienia.

Przykładowa odpowiedź:

```
HTTP 201 OK
{
  "code": 201,
  "message": "Transaction updated successfully",
}
```

Wywołanie operacji anulowania potwierdzonej transakcji (status COMPLETED) zostanie odrzucone z błędem 409 Conflict.

Prawidłowe przetworzenie zmiany statusu płatności ACCEPTED zostanie dodatkowo potwierdzone odpowiednim komunikatem w notyfikacji przesłanej na adres `notifyUrl`. Ewentualne pobrane środki zostaną zwrócone na konto Kupującego.

Anulowanie draftów odbywa się w sposób analogiczny jak anulowanie płatności. Po otrzymaniu żądania PayPo po swojej stronie dokonuje weryfikacji, czy przesłane żądanie dotyczy płatności, czy draftu.

8.2 Zwroty

Zwrot części lub całości środków za zamówienie jest możliwy dla transakcji, które zostały zaakceptowane (status ACCEPTED lub COMPLETED). Ewentualne pobrane środki zostaną zwrócone na konto Kupującego. W przypadku realizacji zwrotu dla

transakcji w statusie ACCEPTED nastąpi jego automatyczne potwierdzenie, status przyjmie wartość COMPLETED.

Dla każdego żądania zwrotu Sprzedawca powinien dodać identyfikator zwrotu (referenceRefundId).

Adres URL: [https://api.paypo.pl/v3/transactions/\\${UUID}/refunds](https://api.paypo.pl/v3/transactions/${UUID}/refunds)

Parametry wejściowe (HTTP POST):

Nazwa parametru	Typ	Wymagany	Opis
amount	int	T	Kwota zwrotu określona w podstawowych jednostkach monetarnych, np. groszach
referenceRefundId	string	N	Identyfikator zwrotu

Ze względu na specyfikę usługi odroczonej płatności nie jest możliwe podwyższenie wartości zamówienia.

Możliwa jest realizacja kilku następujących po sobie zwrotów częściowych pod warunkiem, że suma przesłanych kwot zwrotu nie przekroczy wartości zamówienia. W przypadku zwrotu całości zakupu atrybut amount powinien zawierać aktualną wartość zamówienia.

Przykładowe żądanie dla zwrotu:

```
curl -X POST https://api.paypo.pl/v3/transactions/cd975bc6-a755-4141-b7a0-d7e8f7a308ef/refunds \
-H "Content-Type: application/json" \
-H "Authorization: Bearer fLEsy5wk9jQKK0NjbF57sbOwa4BdCPulc \
-d '{"amount": 24900}'
```

Przykładowa odpowiedź:

```
HTTP 201 Created
{
  "code": 201,
  "message": "Refund created successfully",
}
```

Analogicznie jak w przypadku zmiany statusu, prawidłowe przetworzenie zwrotu zostanie potwierdzone w notyfikacji ze zaktualizowaną kwotą zamówienia przesłanej na adres notifyUrl.

9. Uwierzytelnienie

PayPo API wymaga uwierzytelnienia komunikacji dla metod HTTP modyfikujących transakcje - POST i PATCH - oraz żądań HTTP GET.

Wartości atrybutów wymaganych do podpisywania wiadomości: identyfikatora sprzedawcy `merchantId`, unikalnego klucza `merchantApiKey` oraz OAuth `ClientId` i `ClientSecret` są dostępne z poziomu panelu transakcyjnego PayPo.

9.1 OAuth2

Wymaganą metodą uwierzytelnienia żądań z systemu Sprzedawcy jest standard OAuth w wersji 2.0. PayPo implementuje metodę zgodnie ze specyfikacją protokołu <https://oauth.net/2/>.

W celu pobrania tokenu uwierzytelniającego należy wywołać metodę API `/oauth/tokens`. Zwrócony ciąg jest zakodowany zgodnie ze standardem JWT (<https://tools.ietf.org/html/rfc7519>). Domyślnie token ważny jest przez 30 minut.

Przykładowe żądanie

```
curl -X POST https://api.paypo.pl/v3/oauth/tokens \
-H "Accept: application/json" \
-H "Content-Type: application/x-www-form-urlencoded" \
--data-raw
"grant_type=client_credentials&client_id=${ClientId}&client_secret=${ClientSecret}"
```

Przykładowa odpowiedź:

```
HTTP 200 OK
{
  "token_type": "Bearer",
  "expires_in": 1800,
  "access_token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIwMDRhZjZjNmOC1mMjcZLTRLZjItOWFhOS1jOWI0MDRjNzJnN2IiLCJqdGkiOiJkMmY1MzQ3M2M5ZWMyYjExYmZhN2Q1YTlmMzQ4MTE5YzY2MjY2NjZGFkOWMwZT1lZjVlOTZlMDhkYWI3ZGFjMTljMzA1MWRmNGU2M2JhZjQwNiIsIm1hdCI6MTYwMzk2ODM1MCwibmJmIjo6NjAzOTY4MzUwLCJleHAiOiE2MDM5NzAxNTAsInN1YiI6IiIsInNjb3BlcyI6WyJtZXJjaGFudCJdfQ.V9YkODDb4EbmSp-8q_yv6vS0Q5-Vii43Hi8BH9toA8ISKjqeyhyojCS_qDMVyPGXiOZIBi_oDHNzpXtWBUoiGP1EfYQs7NXIZhrlG72h9-nbssuG25pycUCoF2SdGLfoGr6_uHF5ODokKy6XFE2ogW0q9pqQByYPOVNyvjXG-AYkxAURNK0117M3oA9x2Fr1Vz95rJnVJsgodpi-kojyYlONASJmlpXrhhAxtAE5na6mgrmx7SBsLpNmDgl_z4M9wfr1gRTwIUdqKD8yAxOV9m1Yzt8TgEhzLK2n-OaBzMwCKqPpWF0VnybckCUUyfuFAshQl3CE80lYXSDQqKUWiw"
}
```

9.2 HMAC

Schemat uwierzytelnienia ze skrótem wiadomości przekazywanym w nagłówku jest wykorzystywany w notyfikacjach. Do wyliczenia wartości skrótu wykorzystywane są poniższe dane połączone znakiem „+”:

- metoda HTTP: POST
- URL_PATH (bez części domenowej) endpointu, np. /notifyUrl/{referenceId}
- treść komunikatu w formacie JSON

Przykładowy kod wyliczający skrót w PHP:

```
$json = json_encode($payload, JSON_UNESCAPED_SLASHES |  
JSON_UNESCAPED_UNICODE);  
// „POST+/notifyUrl/{referenceId}+{...}”  
$data = $method . "+" . $endpoint . "+" . $json;  
$hash = base64_encode(hash_hmac('sha256', $data, $merchant_api_key,  
true));  
$data = $method . "+" . parse_url('https://caly-url.pl/notifyUrl',  
PHP_URL_PATH) . "+" . $json;
```

Weryfikacja skrótu po stronie systemu Sprzedawcy nie jest wymagana, ale zalecana ze względów bezpieczeństwa.

10. Statusy operacji

Statusy HTTP

W odpowiedzi na każde zapytanie zwracany jest status HTTP informujący o wyniku przetwarzania żądania.

200 OK – prawidłowe przyjęcie komunikatu

201 Created – prawidłowe utworzenie zasobu

400 Bad Request – niepoprawne dane lub format komunikatu

401 Unauthorized – błąd uwierzytelnienia, np. nieważny token

403 Forbidden – brak uprawnień do zasobu

404 Not Found – URL lub zasób nie został odnaleziony

405 Method Not Allowed – operacja nie jest możliwa dla danego zasobu

409 Conflict – konflikt identyfikacji zasobu (przykładowo numer zamówienia PayPal i numer zamówienia Sprzedającego identyfikują różne obiekty)

503 Service Unavailable – usługa tymczasowo niedostępna

Wywołania API zwracają obiekt JSON zarówno w przypadku sukcesu jak i dla błędów przetwarzania. Zwracany kod odpowiedzi HTTP jest podstawowym wyznacznikiem statusu operacji. Dodatkowo w przypadku błędu 400 zostaje zwrócona szczegółowa informacja zwrotna (errors). Poniżej znajdują się przykładowe odpowiedzi dotyczące błędu 400.

Przykładowa odpowiedź w przypadku błędu:

- w adresie rozliczeniowym,

```
HTTP 400
{
  "code": 400,
  "message": "Bad request",
  "errors": {
    { "path": „billingAddress.zip”,
      "message" "Missing mandatory parameter" },
  }
}
```

- w danych zamówienia.

```
HTTP 400
{
  "code": 400,
  "message": "Bad request",
  "errors": {
    { "path": "order.referenceId",
      "message": "This value should not be blank." },
  }
}
```