

Dokumentacja techniczna integracji z systemem transakcyjnym Axepta

Dokument zawiera informacje na temat RESTful API systemu płatności Axepta.

1. Zarys techniczny

Komunikacja odbywa się ona poprzez wymianę informacji zapisanych w formacie JSON. Każde zapytanie powinno zawierać odpowiednią metodę autoryzacji.

1.1. Statusy transakcji

Transakcje mogą przyjmować następujące statusy:

Status	Opis
new	Nowa, nieobsłużona transakcja
authorized	Autoryzacja transakcji
pending	Oczekiwanie na status
submitted	Wysłana do realizacji
rejected	Transakcja odrzucona
settled	Transakcja zrealizowana
error	Błąd w transakcji
cancelled	Transakcja anulowana

1.2. Metody autoryzacji

Integracja pozwala na autoryzację za pomocą tokenu autoryzacyjnego. Token można znaleźć w panelu administracyjnym Axepta, w części **Ustawienia** a następnie w zakładce **Klucze API**.

Token autoryzacyjny

Aby dokonać autoryzacji zapytania należy w nagłówkach żądania do serwera umieścić dane autoryzacyjne:

```
Accept: application/json
Content-Type: application/json
Authorization: Bearer {token}
```

Kody odpowiedzi HTTP:

Kod HTTP	Znaczenie
----------	-----------

Kod HTTP	Znaczenie
200	Autoryzacja poprawna
401	Nieautoryzowany dostęp, żądanie zasobu, który wymaga uwierzytelnienia
500	Błąd serwera

2. Metody RESTful API

Produkcyjny URL: <https://api.axepta.pl/>

Każdy poprawny adres składa się z trzech części:

- adresu bazowego zapytania (<https://api.axepta.pl/v1/>),
- identyfikatora klienta ([/merchant/{merchantId}](#)),
- funkcji jednoznacznie określającej zakres danych, których dotyczy zapytanie (np. [/transaction](#) lub [/payment](#)).

Każde zapytanie do serwera powinno zawierać dane autoryzacyjne w nagłówkach (**Token autoryzacyjny**).

Endpoint	Zastosowanie	Metoda
/{merchantId}/transaction	Tworzy nową transakcję	POST
/{merchantId}/transaction/{transactionId}	Pobiera dane transakcji	GET
/{merchantId}/payment-link	Tworzy nowy link płatności	POST
/{merchantId}/payment/{paymentId}	Pobiera dane płatności	GET
/{merchantId}/transaction/profile	Obciąża istniejący profil kartowy	POST
/{merchantId}/profile/cid/{cid}	Pobiera profile o podanym cid	GET
/{merchantId}/profile/id/{paymentProfileId}	Pobiera profil o podanym id	GET
/{merchantId}/profile/deactivate	Dezaktywuje profil o podanym w żądaniu identyfikatorze	POST
/{merchantId}/profile/id/{paymentProfileId}	Dezaktywuje profil o podanym paymentProfileId	DELETE
/{merchantId}/payment/cancel	Anuluje link płatności	POST

gdzie:

- [merchantId](#) - identyfikator klienta,
- [serviceId](#) - identyfikator sklepu z którym związane są określone metody realizacji transakcji (UUID v4),
- [cid](#) - identyfikator klienta/płatnika nadany przez akceptanta,
- [transactionId](#) - unikalny identyfikator transakcji (UUID v4),
- [paymentProfileId](#) - identyfikator profilu kartowego,
- [paymentId](#) - unikalny identyfikator linku płatności (UUID v4).

3. Tworzenie nowej transakcji przez API

Zgodnie z wymaganiami PCI DSS (ustawionymi przez organizacje płatnicze) zabronione jest przetwarzanie, przekazywanie czy przechowywanie numerów i innych danych dotyczących kart płatniczych czy kredytowych. Jeśli posiadasz właściwy certyfikat PCI DSS i chcesz udostępnić formatkę płatności kartami na stronie Twojego sklepu - prosimy o kontakt obsługą.

W celu utworzenia nowego zamówienia należy za pomocą metody **POST** przesłać komunikat na endpoint API <https://api.axepta.pl/v1/merchant/{merchantId}/transaction> zawierający informacje o nowym zamówieniu.

gdzie:

- **merchantId** - identyfikator klienta.

3.1. HTTP Request dla płatności PBL

Przykładowy adres na który należy wysłać żądanie POST

```
https://api.axepta.pl/v1/merchant/6yt3ggtm9p7b8h9xsdqz/transaction
```

Nagłówki zapytania

```
Accept: application/json
Authorization: Bearer ad8y3hdoashco8fh49fhiahfb237f8hoihsd0f2hfikljf023h8
Content-Length: 895
Content-Type: application/json
```

Payload zapytania

```
{
  "type": "sale",
  "serviceId": "62f574ed-d4ad-4a7e-9981-89ed7284aaba",
  "amount": 100,
  "currency": "PLN",
  "orderId": "123456789",
  "paymentMethod": "pbl",
  "paymentMethodChannel": "bnpparibas",
  "successReturnUrl": "https://domain.com/success",
  "failureReturnUrl": "https://domain.com/failure",
  "customer": {
    "firstName": "Jan",
    "lastName": "Kowalski",
    "cid": "123",
    "email": "jan.kowalski@example.com"
  }
}
```

```
}  
}
```

Parametry payload

Parametr	Typ	Wymagany	Opis
type	string	✓	Typ transakcji. Dopuszczalne wartości: <code>sale</code> , <code>refund</code> .
serviceId	string(36)	✓	identyfikator sklepu jako UUID v4.
amount	integer	✓	Kwota transakcji w najmniejszej jednostce waluty np. grosze.
currency	string(3)	✓	Waluta transakcji w standardzie ISO 4217.
orderId	string(100)	✓	Numer zamówienia akceptanta, dopuszczalne znaki: A-Za-z0-9#_-. / oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
title	string(255)	-	Tytuł zamówienia, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
paymentMethod	string	✓	Metoda realizacji zamówienia. Dopuszczalne są wartości: <code>pbl</code> , <code>card</code> , <code>blik</code> , <code>bnp-instalments</code> .
paymentMethodChannel	string	✓	Oznaczenie kanału płatności.
successReturnUrl	string(300)	✓	Adres powrotu z zewnętrznej strony obsługującej płatność w przypadku dokonania płatności z powodzeniem.
failureReturnUrl	string(300)	✓	Adres powrotu z zewnętrznej strony obsługującej płatność w przypadku wystąpienia błędu płatności.
returnUrl	string(300)	✓	Domyślny adres powrotu.
customer	object	✓	Dane klienta.
billing	object	-	Dane płatnika.
shipping	object	-	Dane dot. dostawy.
card	object	-	Dane dot. płatności kartą. (Wymagane podczas płatności <code>card</code>)
additionalData	object	-	Informacje o przeglądarce płatnika, wymagane podczas wykonywania płatności kartą z autoryzacją 3ds.

Parametr	Typ	Wymagany	Opis
<code>activeTo</code>	string	-	Data ważności transakcji jako timestamp w sekundach, jeżeli nie jest przekazane to transakcja ważna jest zawsze. Brak realizacji płatności do tego czasu spowoduje jej anulowanie.

Jeżeli w zapytaniu wystąpi parametr `customer`, `billing`, `shipping` lub `card` konieczne jest dostarczenie parametrów:

customer

Parametr	Typ	Wymagany	Opis
<code>firstName</code>	string(100)	✓	Imię klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
<code>lastName</code>	string(100)	✓	Nazwisko klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
<code>cid</code>	string(100)	✓	Identyfikator klienta, dopuszczalne znaki: A-Za-z0-9 oraz myślnik (0x2D)
<code>phone</code>	string(20)	-	Numer telefonu, dopuszczalne znaki: 0-9+- oraz znak spacji (0x20).
<code>email</code>	string(200)	✓	Adres email.
<code>locale</code>	string(2)	-	Lokalizacja płatnika, określa język wysyłanej wiadomości mailowej, informującej o rozpoczęciu procesu płatności, dopuszczalne wartości: <code>pl</code> , <code>en</code> , <code>ua</code> , <code>de</code> .

Domyślnym językiem wysyłanej wiadomości mailowej jest język polski dla transakcji w PLN, oraz język angielski dla pozostałych walut.

billing

Parametr	Typ	Wymagany	Opis
<code>firstName</code>	string(100)	✓	Imię klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)

Parametr	Typ	Wymagany	Opis
lastName	string(100)	✓	Nazwisko klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
company	string(200)	-	Nazwa firmy, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).
street	string(200)	-	Ulica, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).
city	string(100)	-	Miasto, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).
region	string(100)	-	Region, Dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20)
postalCode	string(30)	-	Kod pocztowy.
countryCodeAlpha2	string(20)	-	Kod kraju Alpha2.

shipping

Parametr	Typ	Wymagany	Opis
firstName	string(100)	✓	Imię klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
lastName	string(100)	✓	Nazwisko klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
company	string(200)	-	Nazwa firmy, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).
street	string(200)	-	Ulica, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).
city	string(100)	-	Miasto, dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20).

Parametr	Typ	Wymagany	Opis
region	string(100)	-	Region, Dopuszczalne znaki: A-Za-z0-9 oraz znak spacji (0x20)
postalCode	string(30)	-	Kod pocztowy.
countryCodeAlpha2	string(2)	-	Kod kraju Alpha2.

card

Parametr	Typ	Wymagany	Opis
firstName	string(100)	✓	Imię właściciela karty, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
lastName	string(100)	✓	Nazwisko właściciela karty, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
number	string(16)	✓	Numer karty
month	string(2)	✓	Ważność karty - miesiąc
year	string(4)	✓	Ważność karty - rok
cvv	string(4)	✓	Kod cvv karty

3.2. Metody i kanały realizacji transakcji

Transakcje można realizować następującymi metodami:

Nazwa	Wartość parametru paymentMethod	Wartość parametru paymentMethodChannel
Przelewy online.	pbl	Tabela poniżej
Płatność kart.	card	Tabela poniżej
Płatność BLIK.	blik	blik
Raty BNP.	bnp_instalments	bnp_instalments

Przelew online Pay By Link

Przelewy Pay By Link można realizować za pomocą następujących usług banków:

Wartość parametru	Nazwa usługi
bnpparibas	Płace z BGŻ BNP Paribas

Wartość parametru	Nazwa usługi
mtransfer	mTransfer - mBank
bzwbk	Przelew24 - Santander Bank Polska
pekao24	Pekao24Przelew - Bank Pekao
inteligo	Płacę z Inteligo
ing	Płać z ING
ipko	Płać z iPKO
getin	Płacę z Getin Bank
noble	Płacę z Noble Bank
ideabank	Płacę z Idea Bank - IdeaBank
creditagricole	Credit Agricole e-przelew
alior	Płacę z Alior Bankiem
pbs	Bank Nowy BFG
millennium	Millennium - płatności internetowe
citi	Przelew z Citi Handlowego
bos	Płać z BOŚ
pocztowy	Pocztowy24
plusbank	Płacę z Plus Bank
bs	Bank Spółdzielczy
bspb	Bank Spółdzielczy w Brodnicy
nest	nestPrzelew
envelo	Envelo Bank

Płatność kartą

Płatność kartą można realizować za pomocą następujących usług:

Wartość parametru	Nazwa usługi
ecom3ds	Płatność kartą 3DS
oneclick	Płatność za pomocą usługi oneclick
recurring	Płatność za pomocą usługi recurring

Dane przeglądarki

W przypadku generowania płatności kartą, która wymaga autoryzacji metodą 3D Secure, należy pobrać i przekazać informacje na temat przeglądarki płatnika w dodatkowym obiekcie `additionalData`.

Struktura wygląda w następujący sposób:

Parametr	Typ	Wymagany	Opis
<code>browser</code>	object	Tak	Informacje na temat przeglądarki płatnika
<code>browser.ip</code>	string	Tak	Adres IP. Dopuszczalne wartości: "ipv4", "ipv6"
<code>browser.language</code>	string	Tak	Język przeglądarki określony w formacie UNICODE np. "pl-PL"
<code>browser.jsEnabled</code>	boolean	Tak	Określa, czy w przeglądarce włączona jest obsługa JavaScript
<code>browser.timezoneOffset</code>	integer	Tak	Informacja o strefie czasowej
<code>browser.userAgent</code>	string	Tak	Informacja o używanej aplikacji klienckiej
<code>browser.accept</code>	string	Tak	Akceptowane wartości przez aplikację
<code>browser.javaEnabled</code>	boolean	Tak	Określa, czy w przeglądarce włączona jest obsługa Java
<code>browser.screenColorDepth</code>	integer	Tak	Głębina wyświetlanych kolorów
<code>browser.screenHeight</code>	integer	Tak	Wysokość okna przeglądarki
<code>browser.screenWidth</code>	integer	Tak	Szerokość okna przeglądarki

Przykład przekazywanego obiektu

```
"additionalData": {
  "browser": {
    "ip": "127.0.0.1",
    "language": "pl-PL",
    "jsEnabled": true,
    "timezoneOffset": 100,
    "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome 87.0.4280.66 Safari/537.36",
    "accept": "application/json, text/javascript, */*; q=0.01",
    "javaEnabled": false,
    "screenColorDepth": 24,
    "screenHeight": 1080,
    "screenWidth": 2560
  }
}
```

3.3. HTTP Response

Odpowiedź serwera

W przypadku wykonania poprawnego zapytania rejestrującego nowe zamówienie serwer odpowie statusem HTTP 200 oraz informacją o nowo utworzonej transakcji:

```
{
  "status": "SUCCESS",
  "data": {
    "transaction": {
      "id": "f115d23d-a943-4585-a3d7-09f6c417200d",
      "type": "sale",
      "status": "new",
      "source": "api",
      "createdAt": 1631538444,
      "modifiedAt": 1631538444,
      "notificationUrl": "https://notification_url",
      "serviceId": "63f574ed-d4ad-407e-9981-39ed7584a7b7",
      "amount": 100,
      "currency": "PLN",
      "orderId": "123456789",
      "paymentMethod": "pbl",
      "paymentMethodChannel": "bnpparibas",
      "payment": {
        "id": "bdec4256-618a-48a4-bb0b-a25974654b84",
        "status": "new"
      }
    },
    "action": {
      "type": "redirect",
      "url": "https://tpp.goonline.bnpparibas.pl?token=25667491-0e14-4338-ae71-2aa2c8cbe0ed&state=bdec4256-618a-48a4-bb0b-a25974654b84&requestToken=25667491-0e14-4338-ae71-2aa2c8cbe0ed",
      "method": "GET",
      "contentType": "",
      "contentBodyRaw": ""
    }
  }
}
```

W odpowiedzi otrzymujemy obiekty: **transaction** oraz **action**.

Obiekt **transaction** jest identyczny z wysłanym w zapytaniu rejestrującym zamówienie i zawiera kilka dodatkowych parametrów:

Parametr	Typ	Opis
id	string	Identyfikator transakcji w formacie UUID v4. Unikalny dla każdego zamówienia.
status	string	Status zamówienia.
source	string	Źródło zamówienia. Może posiadać wartości: api lub web .
created	integer	Data utworzenia zamówienia w formacie UNIX TIMESTAMP czasu UTC.

Parametr	Typ	Opis
<code>notificationUrl</code>	string	Adres notyfikacji
<code>modified</code>	integer	Data ostatniej zmiany statusu transakcji w formacji UNIX TIMESTAMP czasu UTC.
<code>payment</code>	object	Dane płatności do której przypisana jest transakcja

Drugim dodatkowym obiektem jest `action`. Obiekt ten wystąpi tylko w przypadku konieczności przekierowania płatnika na zewnętrzną stronę jak to ma miejsce w przypadku płatności `Pay-By-Link`. Obiekt ten zawiera dodatkowe pola których znaczenie jest opisane poniżej:

Parametr	Typ	Opis
<code>type</code>	string	Typ akcji.
<code>url</code>	string	W przypadku konieczności wykonania przekierowania płatnika na inną stronę (np. banku) adres URL.
<code>method</code>	string	Metoda POST lub GET.
<code>contentType</code>	string	Pozycja w nagłówku zapytania do banku określająca typ payloadu.
<code>contentBodyRaw</code>	string	Payload zapytania.

Statusy HTTP

Kod HTTP	Znaczenie
200	Zapytanie wykonane poprawnie. Utworzono transakcję
400	Błędne żądanie, niepoprawny payload żądania.
401	Nieautoryzowany dostęp. Żądanie zasobu, który wymaga uwierzytelnienia.
403	Brak uprawnień do wykonania żądania.
404	Nieznany zasób.
422	Payload jest poprawny ale nie zawiera wymaganych parametrów.
500	Błąd serwera.
503	System niedostępny.

4. Wykonanie zwrotu

Podczas wykonywania wielu transakcji zwrotu jednocześnie, należy wprowadzić co najmniej 5-cio sekundowe opóźnienie pomiędzy kolejnymi transakcjami.

Zwrot wykonywany jest dla najwyższej wartości transakcji przypisanej do danej płatności. Akcję można wykonać, zanim dana płatność zostanie w całości zrealizowana.

Poprawne wykonanie zwrotu polega na wysłaniu żądania metodą POST na adres:

```
https://api.axepta.pl/v1/merchant/{merchantId}/payment/{paymentId}/refund
```

gdzie:

- **merchantId** - identyfikator klienta,
- **paymentId** - unikalny identyfikator płatności której dotyczy zwrot.

W zawartości żądania należy wprowadzić:

```
{
  "type": "refund",
  "serviceId": "24737aab-a507-4feb-8248-3f42bfdbb006",
  "amount": 100
}
```

gdzie:

- **serviceId** - identyfikator sklepu klienta,
- **amount** - kwota do zwrotu.

W żądaniu powinny być również zawarte nagłówki:

```
Content-Type: application/json
Authorization: Bearer {token}
```

5. Tworzenie nowego linku płatności przez API

W celu utworzenia nowego linku płatności należy za pomocą metody **POST** przesłać komunikat na endpoint API **https://api.axepta.pl/v1/merchant/{merchantId}/payment-link** zawierający informacje o nowym linku płatności.

gdzie:

- **merchantId** - identyfikator klienta.

5.1. HTTP Request

Przykładowy adres na który należy wysłać żądanie POST

```
https://api.axepta.pl/v1/merchant/6yt3gjt9p7b8h9xsdqz/payment-link
```

Nagłówki zapytania

```
Accept: application/json
Authorization: Bearer sMkJhaus0ksa87Hbagt+8salsnsJjayPmznx
Content-Type: application/json
Cache-Control: no-cache
```

Payload zapytania

```
{
  "serviceId": "62f574ed-d4ad-4a7e-9981-89ed7284aaba",
  "amount": 100,
  "currency": "PLN",
  "orderId": "123123123",
  "returnUrl": "https://domain.com/return",
  "successReturnUrl": "https://domain.com/success",
  "failureReturnUrl": "https://domain.com/failure",
  "customer": {
    "firstName": "Jan",
    "lastName": "Kowalski",
    "email": "jan.kowalski@example.com",
    "phone": "501501501"
  }
}
```

Parametry payload

Parametr	Typ	Wymagany	Opis
serviceId	string(36)	✓	identyfikator sklepu jako UUID v4.
amount	integer	✓	Kwota transakcji w najmniejszej jednostce waluty np. grosze.
currency	string(3)	✓	Waluta transakcji w standardzie ISO 4217.
orderId	string(100)	✓	Numer zamówienia akceptanta, dopuszczalne znaki: A-Za-z0-9#_-. / oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
title	string(255)	-	Tytuł zamówienia, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)

Parametr	Typ	Wymagany	Opis
<code>returnUrl</code>	string(300)	-	Adres powrotu z zewnętrznej strony obsługującej płatność w przypadku nie rozstrzygnięcia statusu transakcji.
<code>successReturnUrl</code>	string(300)	-	Adres powrotu z zewnętrznej strony obsługującej płatność w przypadku dokonania płatności z powodzeniem.
<code>failureReturnUrl</code>	string(300)	-	Adres powrotu z zewnętrznej strony obsługującej płatność w przypadku wystąpienia błędu płatności.
<code>customer</code>	object	✓	Dane klienta.
<code>activeTo</code>	string	-	Data ważności linku jako timestamp w sekundach, jeżeli nie jest przekazane to link ważny jest zawsze. Brak realizacji płatności do tego czasu spowoduje jej anulowanie.

Parametry dla `customer`:

Parametr	Typ	Wymagany	Opis
<code>firstName</code>	string(100)	✓	Imię klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
<code>lastName</code>	string(100)	✓	Nazwisko klienta, dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
<code>email</code>	string(200)	✓	Adres email.
<code>phone</code>	string(20)	-	Numer telefonu, dopuszczalne znaki: 0-9+- oraz znak spacji (0x20).
<code>cid</code>	string(100)	✓	Identyfikator klienta. dopuszczalne znaki: A-Za-z0-9 oraz myślnik (0x2D)

5.2. HTTP Response

Odpowiedź serwera

W przypadku wykonania poprawnego zapytania rejestrującego nowy link płatności serwer odpowie statusem HTTP **200** oraz informacją o nowo utworzonym linku płatności:

```
{
  "status": "SUCCESS",
```

```

    "data": {
      "paymentLink": {
        "paymentId": "925767db-962a-49ee-916e-783de9b62a73",
        "url": "https://paywall.axepta.pl/pay/925767db-962a-49ee-916e-783de9b62a73"
      }
    }
  }
}

```

W odpowiedzi otrzymujemy obiekt **status** informujący o powodzeniu operacji, oraz **data.paymentLink** z zawartością:

Parametr	Typ	Opis
paymentId	string	Identyfikator płatności w formacie UUID v4. Unikalny dla każdego zamówienia.
url	string	Adres URL linku płatności.

Statusy HTTP

Kod HTTP	Znaczenie
200	Zapytanie wykonane poprawnie. Utworzono transakcję
400	Błędne żądanie, niepoprawny payload żądania.
401	Nieautoryzowany dostęp. Żądanie zasobu, który wymaga uwierzytelnienia.
403	Brak uprawnień do wykonania żądania.
404	Nieznany zasób.
422	Payload jest poprawny ale nie zawiera wymaganych parametrów.
500	Błąd serwera.
503	System niedostępny.

5.3. Multiwypłaty

Opcja dostępna w przypadku włączonej funkcji multiwypłaty. Tworzymy link płatności zgodnie z opisem punktu 5. z jednym dodatkowym parametrem w obiekcie **multipayout**:

- multipayout** - tablica, której każdy element powinien zawierać poniższe pola:
 - ban** - numer konta bankowego,
 - amount** - kwota transakcji podana w groszach,
 - label** - nazwa odbiorcy (max 35 znaków), dopuszczalne znaki: A-Za-z0-9-"'.,/ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne),
 - title** - tytuł przelewu (max 105 znaków), dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne), parametr opcjonalny. Jego obecność powoduje wyodrębnienie danej transakcji na koncie odbiorcy.

Podanie parametru dla jednego elementu warunkuje konieczność dostarczenia go dla pozostałych.

```
{
  "multipayout": [
    {
      "ban": "55105000026800208114085773",
      "amount": 100,
      "label": "Nazwa firmy 0"
    },
    {
      "ban": "55105000026800208114085773",
      "amount": 200,
      "label": "Nazwa firmy 1"
    }
  ]
}
```

6. Raty BNP

Należy utworzyć nową transakcję zgodnie z punktem 3. [Tworzenie nowej transakcji przez API](#) dodając dodatkowy parametr `bnpInstallments`, będący odpowiednio przygotowanymi danymi koszyka.

6.1. Struktura danych

Przykładowa struktura żądania:

```
{
  "bnpInstallments": {
    "address": {
      "street": "Warszawska 1/2",
      "postalCode": "00-000",
      "city": "Miasto"
    },
    "creditTypes": [
      "123",
      "456",
      "789"
    ],
    "products": [
      {
        "name": "bluza",
        "price": 4999,
        "quantity": 2,
        "code": "123"
      }
    ]
  }
}
```



```
}  
}
```

gdzie:

Nazwa	Wym.	Typ	Opis
<code>address</code>	Tak	object	Adres płatnika
<code>address.street</code>	Tak	string	Ulica oraz numer budynku/mieszkania dopuszczalne znaki: A-Za-z0-9-., oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne), przykład: "Ulica 1/2"
<code>address.postalCode</code>	Tak	string	Kod pocztowy, przykład: "00-000"
<code>address.city</code>	Tak	string	Miasto, przykład: "Warszawa"
<code>products</code>	Tak	array	Zamówione produkty
<code>products[].name</code>	Tak	string	Nazwa produktu, dopuszczalne znaki: A-Za-z0-9-., oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne), przykład: "Koszulka"
<code>products[].price</code>	Tak	integer	Cena za pojedynczy produkt podana w groszach, przykład: 4999
<code>products[].quantity</code>	Tak	integer	Liczba zamówionych produktów, przykład: 2
<code>products[].code</code>	Tak	integer	ID produktu, przykład: 123

7. Notyfikacje

Aby poprawnie skonfigurować wysyłkę notyfikacji odnośnie transakcji do sklepu, należy wprowadzić url prowadzący do weryfikacji w panelu administracyjnym Axepta.

W momencie zmiany statusu transakcji serwery Axepta wysyłają powiadomienia na wskazany przez akceptanta adres URL. Wymagane jest by serwer akceptanta (np. sklep) odpowiedział statusem **200 OK** z body `{"status": "ok"}`, który będzie oznaczał poprawne odebranie i przetworzenie notyfikacji przez serwer akceptanta.

Jeśli serwer akceptanta zwróci status 200 OK, lub notyfikacja nie zostanie odebrana do czasu zakończenia trwania cyklu wysyłki, serwery Axepta zaprzestają powtórzeń wysyłki notyfikacji.

7.1. Zawartość BODY notyfikacji

Notyfikacje wysyłane są jako obiekt **JSON** metodą **POST** i mają następującą postać:

```
{
  "payment": {
    "id": "c410aa4c-00c1-4111-97af-0d40b7738881",
    "title": "987654",
    "amount": 100,
    "status": "settled",
    "created": 1623194705,
    "orderId": "123456",
    "currency": "PLN",
    "modified": 1623199529,
    "serviceId": "f0f6cd11-af08-431f-a178-f0ba547c6fe5",
    "notificationUrl": "https://domain.com/",
    "transactions": [
      {
        "id": "8d8c9a1a-59e1-4091-96c7-f315b1c99fb0",
        "type": "sale",
        "status": "settled",
        "source": "web",
        "created": 1623199524,
        "modified": 1623199529,
        "notificationUrl": "https://domain.com/",
        "serviceId": "f0f6cd11-af08-431f-a178-f0ba547c6fe5",
        "amount": 100,
        "currency": "PLN",
        "title": "987654",
        "orderId": "123456",
        "paymentMethod": "blik",
        "paymentMethodChannel": "blik"
      }
    ]
  }
}
```

Notyfikacja może składać się z następujących obiektów:

- **payment** - zawiera informacje o utworzonym linku płatności,
- **transaction** - zawiera informacje o transakcjach jeśli zostały utworzone dla danego linku płatności

7.2. Zawartość nagłówków notyfikacji

Dodatkowo w nagłówkach HTTP umieszczane są następujące parametry:

```
Content-Type: application/json; charset=UTF-8
X-Acepta-Signature: merchantid=6yt3gjt9p7b8h9xsdqz;serviceid=63f574ed-d4ad-407e-9981-
```

```
39ed7584a7b7;signature=20cdc8646eb268ea754842bdf0db1df21a2cf0b1c6e3e16e74ef7f7cca8f5450;alg=sha256
```

gdzie:

- **merchantid** - identyfikator klienta,
- **serviceid** - identyfikator sklepu,
- **signature** - podpis notyfikacji,
- **alg** - algorytm funkcji skrótu (możliwe wartości: **sha256**).

Weryfikacja podpisu notyfikacji jest krytycznym elementem uwierzytelnienia informacji przesyłanych w pakiecie notyfikacji.

7.3. Metoda weryfikacji podpisu notyfikacji

Nagłówek zawierający podpis notyfikacji ma postać:

```
X-axepta-Signature: merchantid=[...];serviceid=[...];signature=[...];alg=[...]
```

Aby uwierzytelnić pochodzenie oraz zweryfikować integralność wiadomości powiadomienia należy wykonać następujące czynności:

1. Z nagłówków pakietu przychodzącego na adres notyfikacji należy pobrać zawartość **X-Axepta-Signature**,
2. Następnie należy pobrać wartość parametru **signature** oraz **alg**,
3. W zależności od algorytmu funkcji skrótu określonego w parametrze **alg** należy obliczyć odpowiednią funkcją skrót:

```
string incoming_signature = x_axepta_signature[signature]
string body = notification_body
string own_signature = hash(body + private_key, alg)
```

4. Obliczoną wartość **own_signature** należy porównać z wartością **incoming_signature**, która została pobrana z nagłówka,
5. Jeżeli wartości **own_signature** i **incoming_signature** są identyczne oznacza to, że wiadomość notyfikacji jest poprawna i pochodzi z zaufanego źródła.

Zmiany statusów transakcji należy dokonywać tylko gdy weryfikacja podpisu przebiegła poprawnie.

8. Pobieranie danych transakcji

Aby pobrać dane zamówienia należy wysłać żądanie **GET** na adres

<https://api.axepta.pl/v1/merchant/{merchantId}/transaction/{transactionId}>.

Przykładowa odpowiedź:

```
{
  "status": "SUCCESS",
  "data": {
    "transaction": {
      "id": "5013e9ff-ece1-452a-9bd9-3b867a952794",
      "type": "sale",
      "status": "pending",
      "source": "api",
      "createdAt": 1631261710,
      "modifiedAt": 1631261710,
      "notificationUrl": "http://domain.com/",
      "serviceId": "33532abf-f5f7-42b7-aa68-d4f93ebad053",
      "amount": 100,
      "currency": "PLN",
      "orderId": "123456",
      "paymentMethod": "pbl",
      "paymentMethodChannel": "bnpparibas",
      "payment": {
        "id": "41f9f9fa-c7d0-4f08-8fb9-007cc24428a6",
        "status": "pending"
      }
    }
  }
}
```

9. Płatność oneclick i rekurencyjna

Niżej wymienione metody płatności są dostępne tylko i wyłącznie dla akceptantów którzy posiadają certyfikat PCI-DSS. Obciążanie karty może odbywać się jedynie na podstawie wyraźnych zgód posiadacza instrumentu płatniczego. Płatnik musi mieć możliwość rezygnacji z subskrypcji.

Płatności **oneclick** polegają na obciążeniu zarejestrowanego już w systemie Axepta profilu klienta/płatnika. Pierwsza płatność wymaga weryfikacji i podania pełnych danych instrumentu płatniczego. Każda kolejna płatność odbywa się za pomocą obciążenia na podstawie przydzielonego do instrumentu płatniczego identyfikatora.

Płatności **rekurencyjne** polegają na cyklicznym obciążaniu instrumentu płatniczego klienta bez weryfikacji. Rejestracja instrumentu płatniczego odbywa się podobnie jak w płatności **oneclick**. Zgodnie z regulacjami organizacji płatniczych płatność musi być powtarzalna co do kwoty oraz okresu czasu.

Dokonując płatności należy pamiętać aby każdy instrument płatniczy posiadał swój unikalny identyfikator klienta/płatnika - **cid**.

9.1. Zarejestrowanie nowego profilu

Rejestracja nowego profilu odbywa się za pomocą utworzenia nowej transakcji (punkt 3.) kartowej z parametrem **paymentMethodChannel** o wartości **oneclick** lub **recurring**. Gdy transakcja zostanie

zaakceptowana, standardowa notyfikacja zostanie zwrócona wzbogacona o dodatkowe pole `statusCode` oraz sekcję `paymentProfile` o strukturze:

```
{
  "paymentProfile": {
    "id": "d6a5bd6c-8e9c-496e-beb2-f0ca08215645",
    "merchantMid": "6yt3gjtm9p7b8h9xsdqz",
    "merchantCustomerId": "123",
    "firstName": "Jan",
    "lastName": "Kowalski",
    "maskedNumber": "****1791",
    "month": "10",
    "year": "2020",
    "organization": "MASTERCARD",
    "isActive": 1,
    "profile": "ONE_CLICK"
  }
}
```

gdzie:

- `id` - identyfikator profilu,
- `merchantMid` - identyfikator klienta,
- `merchantCustomerId` - identyfikator płatnika, dopuszczalne znaki: A-Za-z0-9_-
- `firstName` - imię posiadacza instrumentu płatniczego na który jest zarejestrowany profil,
- `lastName` - imię posiadacza instrumentu płatniczego na który jest zarejestrowany profil,
- `maskedNumber` - cztery gwiazdki oraz ostatnie cztery cyfry instrumentu płatniczego,
- `month` - data ważności karty: miesiąc,
- `year` - data ważności karty: rok,
- `organization` - organizacja płatnicza która wydała zarejestrowaną kartę,
- `isActive` - aktywność profilu: 1 - aktywna, 0 - nieaktywna,
- `profile` - rodzaj profilu.

9.2. Obciążenie istniejącego profilu

Wysyłając żądanie `POST` na adres

<https://api.axepta.pl/v1/merchant/{merchantId}/transaction/profile> możemy obciążyć istniejący profil w systemie.

Payload zapytania powinien zawierać JSON:

```
{
  "serviceId": "63f574ed-d4ad-407e-9981-39ed7584a7b7",
  "paymentProfileId": "39ac1087-e632-41ff-acb8-8d661068a9d5",
  "amount": 100,
  "currency": "PLN",
  "orderId": "123456",
}
```

gdzie:

Parametr	Typ	Wymagany	Opis
serviceId	string(36)	✓	identyfikator sklepu jako UUID v4.
paymentProfileId	string(36)	✓	identyfikator profilu który ma zostać obciążony.
amount	integer	✓	Kwota transakcji w najmniejszej jednostce waluty np. grosze.
currency	string(3)	✓	Waluta transakcji w standardzie ISO 4217.
orderId	string(100)	✓	Numer zamówienia akceptanta - dopuszczalne znaki: A-Za-z0-9#_-./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)
title	string(255)	-	Tytuł transakcji - dopuszczalne znaki: A-Za-z0-9#&_-'",./ oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne)

W przypadku próby obciążenia nieaktywnego profilu odpowiedź będzie wyglądać:

```
{
  "apiErrorResponse": {
    "code": "TRX-ERROR-120301",
    "message": "Payment profile inactive.",
    "instance": {
      "serviceId": "6879ff96-3efa-4ea2-b193-59401da40f18",
      "paymentProfileId": "d6a5bd6c-8e9c-496e-beb2-f0ca08215645",
      "amount": 100,
      "currency": "PLN",
      "orderId": "2171ef23-828e-47e1-a3f6-80fdd4030863"
    },
    "errors": []
  }
}
```

10. Pobieranie danych płatności

Aby pobrać dane płatności należy wysłać żądanie **GET** na adres

<https://api.axepta.pl/v1/merchant/{merchantId}/payment/{paymentId}>.

Przykład odpowiedzi:

```
{
  "status": "SUCCESS",
```

```

    "data": {
      "payment": {
        "id": "1572084b-4b63-40a7-90b1-17199172a385",
        "url": "https://paywall.axepta.pl/pay/1572084b-4b63-40a7-90b1-17199172a385",
        "serviceId": "10af53c6-35ce-4713-81ba-883501277d40",
        "orderId": "123123123",
        "amount": "100",
        "amountPaid": 0,
        "amountRefunded": 0,
        "amountSubmittedRefund": 0,
        "currency": "PLN",
        "status": "new",
        "isActive": true,
        "createdAt": 1631540831,
        "modifiedAt": 1631540831,
        "isGenerated": false,
        "isUsed": false,
        "isConfirmVisited": false,
        "returnUrl": "https://domain.com/return",
        "failureReturnUrl": "https://domain.com/failure",
        "successReturnUrl": "https://domain.com/success",
        "customer": {
          "firstName": "Jan",
          "lastName": "Kowalski",
          "email": "jan.kowalski@example.com",
          "phone": "501501501",
          "locale": "pl",
          "cid": "123"
        }
      }
    }
  }
}

```

11. Minimalne wartości kwot transakcji, zwrotów

Dla każdej metody płatności obowiązują następujące limity:

Metoda płatności	Minimalna kwota płatności (PLN)
Przelewy online Pay-By-Link	1
Płatność za pomocą BLIK	0.10
Płatność kartą	0.05
Płatność oneclick	0.05
Płatność recurring	0.05

Poniżej tego progu dana metoda płatności nie jest dostępna.