

Dokumentacja techniczna integracji sklepu z bramką płatności Acepta

Instrukcja integracji sklepu z bramką płatności Acepta

1. Wykonanie transakcji

Aby wykonać transakcję, należy wysłać żądanie metodą POST na adres:

- produkcja

```
https://paywall.acepta.pl/payment
```

następujące parametry:

Pola wymagane:

- **serviceId** - identyfikator sklepu klienta jako UUID v4,
- **merchantId** - identyfikator klienta,
- **amount** - kwota transakcji podana w groszach,
- **currency** - waluta transakcji w standardzie ISO 4217,
- **orderId** - ID zamówienia,
Dopuszczalne znaki: A-Za-z0-9#_-. / oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne),
- **customer.firstName** - imię osoby składającej zamówienie,
Dopuszczalne znaki: A-Za-z0-9-., oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne),
- **customer.lastName** - nazwisko osoby składającej zamówienie,
Dopuszczalne znaki: A-Za-z0-9-., oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0 (m.in. polskie znaki diakrytyczne),
- **customer.email** - adres e-mail osoby składającej zamówienie,
- **signature** - wyliczona sygnatura.

Pola opcjonalne:

- **customer.phone** - numer telefonu osoby składającej zamówienie,
Dopuszczalne znaki: +-0-9 oraz spacja. Wartość parametru zawierać może maksymalnie 20 znaków.
- **successReturnUrl** - adres URL, na który użytkownik będzie przekierowany po pomyślnie zakończonej transakcji,
- **failureReturnUrl** - adres URL, na który użytkownik będzie przekierowany po błędnie dokonanej transakcji,
- **returnUrl** - adres URL, na który użytkownik będzie przekierowany po dokonaniu transakcji,
- **orderDescription** - tytuł transakcji,
Dopuszczalne znaki: A-Za-z0-9#&_-. / oraz znak spacji(0x20) i znaki z zakresu UNICODE 00C0 - 02C0

(m.in. polskie znaki diakrytyczne),

- `visibleMethod` - metody płatności widoczne na bramce płatności.

Dostępne wartości: card, pbl, blik, bnp_installments.

Jeśli tego pola nie ma lub jest puste to są wyświetlane wszystkie włączone w sklepie klienta metody płatności. Poszczególne parametry należy przekazać jako wartości tablicy

- `cartData` - pole wymagane przy płatnościach ratalnych. Wartością tego parametru powinna być zawartość koszyka, którego przygotowanie opisano w punkcie [4. Raty BNP](#),
- `activeTo` - data ważności linku płatności jako timestamp w sekundach, jeżeli nie jest przekazane to link ważny jest zawsze. Brak realizacji płatności do tego czasu spowoduje jej anulowanie.

Przykładowa zawartość formularza wysyłana metodą POST:

```
<input type="hidden" value="63f574ed-d90d-4abe-9cs1-39117584a7b7"
name="serviceId">
<input type="hidden" value="6yt3gjtm9p1odfgx8491" name="merchantId">
<input type="hidden" value="300" name="amount">
<input type="hidden" value="PLN" name="currency">
<input type="hidden" value="123" name="orderId">
<input type="hidden" value="Example transaction" name="orderDescription">
<input type="hidden" value="John" name="customer[firstName]">
<input type="hidden" value="Doe" name="customer[lastName]">
<input type="hidden" value="johndoe@domain.com" name="customer[email]">
<input type="hidden" value="501501501" name="customer[phone]">
<input type="hidden" value="https://your-shop.com/success"
name="successReturnUrl">
<input type="hidden" value="https://your-shop.com/failure"
name="failureReturnUrl">
<input type="hidden" value="https://your-shop.com/return" name="returnUrl">
<input type="hidden"
value="cd5024f5ce5e6ff47990fe60572758fbcbcd6e3c04895d6815932b2d14e04ffd;sha256"
name="signature">
```

Wartości parametrów z adresami powinny być pełnymi adresami (absolute URL).

1.1. Wyliczanie sygnatury

Sygnaturę wyliczamy w następujący sposób:

1. Sortujemy alfabetycznie, rosnąco, po kluczach parametry zamówienia.
2. Łączymy parametry w następujący sposób:
`parametr1=wartosc1¶metr2=wartosc2...parametrN=wartoscN`. Wynik zapisujemy do zmiennej (zwanej dalej `body`).
3. Wyliczamy sygnaturę metodą hashowania `sha256`: `sha256(body + private_key)`. Wynik hashowania zapisujemy do zmiennej (zwanej dalej `signature`).
4. Do wyliczonej sygnatury dopisujemy po średniku użytą metodę hashowania: `signature + ';sha256'`.

1.1.1. Przykład wyliczenia sygnatury

```

/**
 * @param array $orderData
 * @param string $serviceKey
 * @param string $hashMethod
 *
 * @return string
 */
function createSignature($orderData, $serviceKey, $hashMethod)
{
    $data = prepareData($orderData);

    return hash($hashMethod, $data . $serviceKey);
}

/**
 * @param array $data
 * @param string $prefix
 *
 * @return string
 */
function prepareData(
    $data,
    $prefix = ''
) {
    ksort($data);
    $hashData = [];
    foreach($data as $key => $value) {
        if($prefix) {
            $key = $prefix . '[' . $key . ']';
        }
        if(is_array($value)) {
            $hashData[] = prepareData($value, $key);
        } else {
            $hashData[] = $key . '=' . $value;
        }
    }

    return implode('&', $hashData);
}

$hashMethod = 'sha256';

$serviceKey = 'eAyhFLuHgw15hu-32GM8QV1CVMWRU0dGjH1c';

$fields = [
    'merchantId' => '6yt3gjtm9p1odfgx8491',
    'serviceId' => '63f574ed-d90d-4abe-9cs1-39117584a7b',
    'amount' => '300',
    'currency' => 'PLN',
    'orderId' => '123',
    'orderDescription' => 'Example transaction',
    'customer[firstName]' => 'John',
    'customer[lastName]' => 'Doe',

```

```
'customer[email]' => 'johndoe@domain.com',
'customer[phone]' => '501501501',
'successReturnUrl' => 'https://your-shop.com/success',
'failureReturnUrl' => 'https://your-shop.com/failure',
'returnUrl' => 'https://your-shop.com/return',
'cartData' =>
'H4sIAAAAAAAAAA62QwU7DMAyGX6XyuYi069S1t4kbBcQNCcTBtBGzmiY1SVexam/G1ffCGVC60znZv/3Zf
zwBdmbQHspk1RV5DLWV6GWzDcq6yEW2EcUmBmwaK52DcoIXUor3awg1dhJKuEYdVWZE5VqCGJy3UjIPD2j
dAUfXYpRcplzpjf0orkwTKCEuhBCs1uTf0b81dN6EPsiyQbq/CdD06NC/Fj8PjjE0UtFebqbZxVZrj07YR
rv0UI0kPRfyf1zP+920+v78CtVgSVom5osyEsMe0UxXwT05+rsyM4+aPj9aXEJFUSwh8rLjsz9NQA0DCbf
+7jPuMKgzODvBbwNqf/pU+jfp+fgFAFFF00sBAAA=',
];
$result = createSignature($fields, $serviceKey, $hashMethod) . ';' . $hashMethod;
```

2. Obsługa notyfikacji

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.

2.1. Weryfikacja dostarczonych notyfikacji transakcji

Adres notyfikacyjny można ustawić w panelu administracyjnym systemu Axepta.

Aby upewnić się, że notyfikacja została przysłana z zaufanego źródła, należy dokonać jej weryfikacji. Każda dostarczona notyfikacja, posiada nagłówek **Content-Type** (który rozpoczyna się od: `application/json`) oraz sygnaturę.

Należy zweryfikować czy sygnatura została poprawnie podpisana. Całą zawartość notyfikacji łączymy z kluczem `private_key` i porównujemy wynik z przyslaną sygnaturą w nagłówku **X-Axepta-Signature**. Jeżeli wynik zgadza się - możemy być pewni, że notyfikacja została wysłana z poprawnego źródła.

```
$mySignature = hash($hashMethod, $payload . $serviceKey);
if ($mySignature === $headerSignature) {
// Notyfikacja zweryfikowana poprawnie. Przetwarzaj dalej.
} else {
// Notyfikacja zweryfikowana negatywnie. Ignoruj notyfikację.
}
```

Zmienna `$headerSignature` ma wartość `signature` w nagłówku notyfikacji: **X-Axepta-Signature**, czyli na przykład:

```
X-Acepta-Signature: merchantid=6yt3gjt1234f8h9xsdqz;serviceid=53f574ed-d4ad-aabe-9981-39ed7584a7b7;signature=cd5024f5ce5e6ff47990fe60572758fbcbcd6e3c04895d6815932b2d14e04ffd;sha256
```

2.2. Opis budowy notyfikacji

Kluczowymi nagłówkami są:

```
X-Acepta-Signature: merchantid=6yt3gjt1234f8h9xsdqz;serviceid=53f574ed-d4ad-aabe-9981-39ed7584a7b7;signature=5e2ac79f4f02d368cdd6eae17d2089dec13577c1d6e3364d6fc123c85029e82;alg=sha256
Content-Type: application/json; charset=UTF-8
```

gdzie:

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

Zawartość notyfikacji (body):

```
{
  "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

3. 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}
```

4. Raty BNP

Aby rozpocząć płatność ratalną, należy wykonać standardową transakcję według punktu

[1. Wykonanie transakcji](#), dodając dodatkowy parametr `cartData`.

Parametr ten zawiera spakowane dane koszyka, przygotowane według punktu [4.2](#).

Pole to również należy uwzględnić przy wyliczaniu sygnatury.

4.1. Struktura danych

Przykładowa struktura pakietu przed spakowaniem:

```
{
  "customer": {
    "address": {
      "billing": {
        "street": "Warszawska 1/2",
        "postalCode": "00-000",
        "city": "Miasto",
        "country": "PL",
      },
      "delivery": {
        "street": "Kwietna 7",
        "postalCode": "00-000",
        "city": "Miasto",
        "country": "PL",
      }
    }
  },
  "cart": [
    {
      "name": "Bluza",
      "price": 4999,
      "quantity": 2,
      "code": 123
    }
  ],
  "grandTotal": 9998
}
```

gdzie:

Nazwa	Wym.	Typ	Opis
-------	------	-----	------

Nazwa	Wym.	Typ	Opis
<code>customer</code>	Tak	object	Informacje o płatniku
<code>address</code>	Tak	object	Adresy płatnika
<code>address.billing</code>	Tak	object	Adres rozliczeniowy
<code>address.delivery</code>	Tak	object	Adres dostawy, posiada taki sam zestaw parametrów co <code>address.billing</code>
<code>address.billing.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.billing.postalCode</code>	Tak	string	Kod pocztowy, przykład: "00-000"
<code>address.billing.city</code>	Tak	string	Miasto, przykład: "Warszawa"
<code>address.billing.country</code>	Tak	string	Kraj, przykład: "PL"
<code>cart</code>	Tak	array	Zamówione produkty
<code>cart[].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>cart[].price</code>	Tak	integer	Cena za pojedynczy produkt podana w groszach, przykład: 4999
<code>cart[].quantity</code>	Tak	integer	Liczba zamówionych produktów, przykład: 2
<code>cart[].code</code>	Tak	integer	Identyfikator produktu, przykład: 123
<code>grandTotal</code>	Tak	integer	Całkowita wartość koszyka

4.2. Przygotowanie danych koszyka

Za pomocą funkcji PHP

Zebrane dane należy spakować za pomocą poniższej funkcji:

```
base64_encode(gzencode(json_encode($fieldsData), 5));
```


gdzie:

- `fieldsData` - przygotowane wcześniej pola koszyka według struktury z punktu 4.1.

Za pomocą biblioteki JavaScript

Biblioteka JS wspomagająca proces przygotowania danych koszyka znajduje się na stronie internetowej:

```
https://github.com/nodeca/pako
```

Użycie biblioteki wygląda w następujący sposób:

```
btoa(pako.gzip(JSON.stringify(fieldsData), {to: 'string', level: 5}))
```

gdzie:

- `fieldsData` - przygotowane wcześniej pola koszyka według struktury z punktu 4.1.

4.3. Przykład parametru

Przykładowa zawartość pola `cartData` po spakowaniu:

```
<input type="hidden"
value="H4sIAAAAAAAAAA62QwU7DMAyGX6XyuYi069S1t4kbBcQNCcTBtBGzmiY1SVexam/GlffCGVC60zn
Zv/3ZfzwBdmbQHspk1RV5DLWV6GWzDcq6yEW2EcUmBmwaK52DcoIXUor0awg1dhJKuEYdVWZE5VqCGJy3U
jIPD2jdAUfXYpRcp1zpjf0orkwTKCEuhBCs1uTf0b8ldN6EPBiyQbq/CdD06NC/Fj8PjjE0UtFehqbZxVZ
rj07YRrv0UI0kPRfyf1zP+920+v78CtVgSVom5osyEsMeOUxXwT05+rsyM4+aPj9aXEJFUSwh8rLjsz9NQ
A0DCbf+7jPuMKgzODvBbwNqf/pU+jfp+fgFAFF00sBAAA=" name="cartData">
```

6. Multiwypłaty

Opcja dostępna w przypadku włączonej funkcji multiwypłaty. Wykonujemy transakcje zgodnie z opisem punktu 1. z jednym dodatkowym parametrem:

- `multipayout` - tablic, której każdy element powinien zawierać wszystkie poniższe pola:
 - `iban` - 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.

Każda wypłata zawarta w formularzu poniżej powinna zawierać kolejne indexy numerowane od 0.

Przykładowa zawartość formularza wysłana metodą POST:

```
<input type="hidden" value="63f574ed-d90d-4abe-9cs1-39117584a7b7"
name="serviceId">
<input type="hidden" value="6yt3gjtm9p1odfgx8491" name="merchantId">
<input type="hidden" value="300" name="amount">
<input type="hidden" value="PLN" name="currency">
<input type="hidden" value="123" name="orderId">
<input type="hidden" value="Example transaction" name="orderDescription">
<input type="hidden" value="John" name="customer[first_name]">
<input type="hidden" value="Doe" name="customer[last_name]">
<input type="hidden" value="johndoe@domain.com" name="customer[email]">
<input type="hidden" value="501501501" name="customer[phone]">
<input type="hidden" value="https://your-shop.com/success"
name="successReturnUrl">
<input type="hidden" value="https://your-shop.com/failure"
name="failureReturnUrl">
<input type="hidden" value="https://your-shop.com/return" name="returnUrl">
<input type="hidden"
value="H4sIAAAAAAAAAA62QwU7DMAyGX6XyuYi069S1t4kbBcQNCcTBtBGzmiY1SVexam/GlffCGVC60zn
Zv/3ZfzwBdmbQHspk1RV5DLWV6GWzDcq6yEW2EcUmBmwaK52DcoIXUor0awg1dhJKuEYdVYZE5VqCGJy3U
jIPD2jdAUfXYpRcp1zpjf0orkwTKCEuhBCs1uTf0b8ldN6EPBiyQbq/CdD06NC/Fj8PjjE0UtFehqbZxVZ
rj07YRrv0UI0kPRfyf1zP+920+v78CtVgSVom5osyEsMeOUxXwT05+rsyM4+aPj9aXEJFUSwh8rLjsz9NQ
A0DCbf+7jPuMKgzODvBbwNqf/pU+jfp+fgFAFF00sBAAA=" name="cartData">
<input type="hidden" value="PL7210500028166973380325415" name="multipayout[0]
[iban]">
<input type="hidden" value="100" name="multipayout[0][amount]">
<input type="hidden" value="Nazwa firmy 0" name="multipayout[0][label]">
<input type="hidden" value="PL5810500025503268251444948" name="multipayout[1]
[iban]">
<input type="hidden" value="200" name="multipayout[1][amount]">
<input type="hidden" value="Nazwa firmy 1" name="multipayout[1][label]">
<input type="hidden"
value="1a466af99a18c4691576bbbf5b935e2ac082285ae28a88f8686ac3317836a6f5;sha256"
name="signature">
```

6.1. Wylizanie sygnatury

Sygnature obliczamy zgodnie z punktem 1.1. z uwzględnieniem dodatkowych parametrów

```
$fields = [
  'merchantId' => '6yt3gjtm9p1odfgx8491',
  'serviceId' => '63f574ed-d90d-4abe-9cs1-39117584a7b',
  'amount' => '300',
  'currency' => 'PLN',
  'orderId' => '123',
  'orderDescription' => 'Example transaction',
  'customer[first_name]' => 'John',
  'customer[last_name]' => 'Doe',
  'customer[email]' => 'johndoe@domain.com',
```

```

'customer[phone]' => '501501501',
'successReturnUrl' => 'https://your-shop.com/success',
'failureReturnUrl' => 'https://your-shop.com/failure',
'returnUrl' => 'https://your-shop.com/return',
'cartData' =>
'H4sIAAAAAAAAAA62QwU7DMAyGX6XyuYi069S1t4kbBcQNCcTBtBGzmiYlSVexam/GlffCGVC60znZv/3Zf
zwBdmbQHspklRV5DLWV6GWzDcq6yEW2EcUmBmwaK52DcoIXUor0awg1dhJKuEYdVWZE5VqCGJy3UjIPD2j
dAUfXYpRcplzpjf0orkwTKCEuhBCs1uTf0b8ldN6EPBiyQbq/CdD06NC/Fj8PjjE0UtFehqbZxVZrj07YR
rv0UI0kPRfyf1zP+920+v78CtVgSVom5osyEsMeOUxXwT05+rsyM4+aPj9aXEJFUSwh8rLjsz9NQA0DCbf
+7jPuMKgzODvBbwNqf/pU+jfp+fgFAfFF00sBAAA=',
'multipayout' => [
  [
    'iban' => '72105000028166973380325415',
    'amount' => '100',
    'label' => 'Nazwa firmy 0',
  ],
  [
    'iban' => '58105000025503268251444948',
    'amount' => '200',
    'label' => 'Nazwa firmy 1',
  ],
],
];

```

7. 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

Poniżej tego progu dana metoda płatności nie jest dostępna.