

BlueMedia PHP SDK

Data wygenerowania: 2021-11-23

BlueMedia PHP SDK	3
Wymagania	3
Instalacja	3
Konfiguracja klienta	3
Transakcja poprzez przekierowanie na paywall	3
Przedtransakcja	4
Przedtransakcja, link do kontynuacji płatności	5
Przedtransakcja, brak kontynuacji	5
Szybki przelew	6
Obsługa ITN (Instant Transaction Notification)	7
Obsługa ITN, utworzenie obiektu komunikatu	7
Pobieranie listy aktualnie dostępnych regulaminów	7
Pobieranie listy kanałów płatności	8

BlueMedia PHP SDK

Kod zawarty w tym repozytorium umożliwia wykonanie transakcji oraz innych usług oferowanych przez Blue Media S.A.

Użycie SDK zalecane jest podczas implementacji własnych modułów płatności. Możesz je pobrać [tutaj](#)

Uwaga: w wersji 1.0.0 możliwe jest wykonanie płatności oraz ITN z ograniczonym zestawem parametrów.

Wymagania

- PHP w wersji 7.2 lub nowszej.
- Rozszerzenia PHP:
 - xmlwriter,
 - xmlreader,
 - iconv,
 - mbstring,
 - hash

Instalacja

```
$ composer require bluepayment-plugin/bm-sdk
```

Konfiguracja klienta

W celu utworzenia warstwy komunikacji należy utworzyć obiekt klasy `BlueMedia\Client` podając id serwisu oraz klucz współdzielony (przyznane przez BlueMedia).

```
$client = new BlueMedia\Client('ID SERWISU', 'KLUCZ WSPÓLDZIELONY');
```

Podczas tworzenia obiektu klienta, za argumentami danych serwisu można dodatkowo dodać użyty tryb szyfrowania oraz separator danych (w przypadku kiedy są nadane inne niż domyślne):

```
$client = new BlueMedia\Client(  
    'ID SERWISU',  
    'KLUCZ WSPÓLDZIELONY',  
    'sha256', // tryb hashowania, domyślnie sha256, można użyć stałej z  
    BlueMedia\Common\Enum\ClientEnum  
    '|' // separator danych, domyślnie |  
);
```

Transakcja poprzez przekierowanie na paywall

Najprostszym typem wykonania transakcji jest przekierowanie do serwisu BlueMedia wraz z danymi o transakcji. Obsługa płatności leży wtedy w całości po stronie serwisu BlueMedia.

Aby wykonać transakcję należy wywołać metodę `getTransactionRedirect`, poprawne wykonanie metody zwróci formularz który wykona przekierowanie do serwisu BlueMedia:

```
$result = $client->getTransactionRedirect([
    'gatewayUrl' => 'https://pay-accept.bm.pl', // Adres bramki BlueMedia
    'transaction' => [
        'orderId' => '123', // Id transakcji, wymagany
        'amount' => '1.20', // Kwota transakcji, wymagany
        'description' => 'Transakcja 123-123', // Tytuł transakcji, opcjonalny
        'gatewayID' => '0', // Identyfikator kanału płatności, opcjonalny, w tym
        // przypadku można ustawić jako 0 lub pominąć
        'currency' => 'PLN', // Waluta transakcji, opcjonalny, domyślnie PLN
        'customerEmail' => 'test@hostname.domain' // Email klienta, opcjonalny, zalecany
        // ze względu na automatyczne uzupełnienie pola po stronie serwisu BM
    ]
]);

echo $result->getData();
```

Po wykonaniu płatności, serwis BlueMedia wykona przekierowanie na skonfigurowany wcześniej adres powrotu płatności. Przekierowanie następuje poprzez żądanie HTTPS (GET) z trzema parametrami:

- ServiceID - Identyfikator serwisu
- OrderID - Identyfikator transakcji
- Hash - Suma kontrolna wyliczona na podstawie ServiceID i OrderID.

Wymagane jest, aby strona powrotu z płatności weryfikowała poprawność Hash, służy do tego metoda `doConfirmationCheck`. Należy przekazać do niej dane przesłane w żądaniu GET:

```
$data = [
    'ServiceID' => '123456',
    'OrderID' => '123',
    'Hash' => 'df5f737f48bcef93361f590b460cc633b28f91710a60415527221f9cb90da52a'
];

$result = $client->doConfirmationCheck($data); // true | false
```

Przedtransakcja

Metoda `doTransactionInit` rozszerza standardowy model rozpoczęcia transakcji o obsługę określonych potrzeb:

- zamówienia linku do płatności na podstawie przesłanych parametrów
- obciążenia Klienta (jeśli nie jest wymagana dodatkowa autoryzacja dokonana przez Klienta)
- zweryfikowania poprawności linku płatności, zanim Klient zostanie przekierowany do Systemu - wywołanie powoduje walidację parametrów i konfiguracji Systemu
- skrócenia linka płatności - zamiast kilku/kilkunastu parametrów, link zostaje skrócony do dwóch identyfikatorów

- ukrycia danych wrażliwych parametrów linku transakcji – przedtransakcja odbywa się backendowo, a link do kontynuacji transakcji nie zawiera danych wrażliwych, a jedynie identyfikatory kontynuacji
- użycia SDK w modelu pełnym (bezpiecznym)

Metoda przyjmuje parametry takie jak w przypadku transakcji z przekierowaniem na paywall, z tą różnicą że wysyłany jest inny nagłówek, dzięki czemu serwis BlueMedia obsługuje żądanie w inny sposób. W odpowiedzi otrzymywany jest link do kontynuacji transakcji lub odpowiedź informująca o braku kontynuacji oraz statusem płatności.

Przedtransakcja, link do kontynuacji płatności

```
$result = $client->doTransactionInit([
    'gatewayUrl' => 'https://pay-accept.bm.pl',
    'transaction' => [
        'orderId' => '123',
        'amount' => '1.20',
        'description' => 'Transakcja 123-123',
        'gatewayID' => '0',
        'currency' => 'PLN',
        'customerEmail' => 'test@hostname.domain'
    ]
]);

$transactionContinue = $result->getData();

$transactionContinue->getRedirectUrl(); //
https://pay-accept.bm.pl/payment/continue/9IA2UISN/718GTV5E
$transactionContinue->getStatus(); // PENDING
$transactionContinue->getOrderId(); // 123
$transactionContinue->toArray(); // [...]
// ...
```

Przedtransakcja, brak kontynuacji

```
$result = $client->doTransactionInit([
    'gatewayUrl' => 'https://pay-accept.bm.pl',
    'transaction' => [
        'orderId' => '123',
        'amount' => '1.20',
        'description' => 'Transakcja 123-123',
        'gatewayID' => '1500',
        'currency' => 'PLN',
        'customerEmail' => 'test@hostname.domain',
        'customerIP' => '127.0.0.1',
        'title' => 'Test',
    ]
]);

$transactionInit = $result->getData();

$transactionInit->getConfirmation(); // NOTCONFIRMED
$transactionInit->getReason(); // MULTIPLY_PAID_TRANSACTION
$transactionInit->getOrderId(); // 123
$transactionInit->toArray(); // [...]
// ...
```

Szybki przelew

Szybki Przelew to forma płatności, która wymaga od Klienta samodzielnego przepisania danych do przelewu dostarczanych przez System. Dane do przelewu można pozyskać dzięki metodzie `doTransactionBackground`.

W zależności od kanału płatności jaki zostanie wybrany w kontekście transakcji, metoda zwróci dane do przelewu lub gotowy formularz.

Przykład wywołania (dane do transakcji):

```
$result = $client->doTransactionBackground([
    'gatewayUrl' => 'https://pay-accept.bm.pl',
    'transaction' => [
        'orderID' => '12345',
        'amount' => '5.12',
        'description' => 'Test transaction 12345',
        'gatewayID' => '21',
        'currency' => 'PLN',
        'customerEmail' => 'test@test.test',
        'customerIP' => '127.0.0.1',
        'title' => 'Test',
        'validityTime' => date('Y-m-d H:i:s', strtotime('now +5 hour')),
        'linkValidityTime' => date('Y-m-d H:i:s', strtotime('now +5 hour'))
    ]
]);

$transactionBackground = $result->getData();

$transactionBackground->getReceiverNRB(); // 47 1050 1764 1000 0023 2741 0516
$transactionBackground->getReceiverName(); // Blue Media
$transactionBackground->getBankHref(); // https://ssl.bsk.com.pl/bskonl/login.html
$transactionBackground->toArray(); // [...]
// ...
```

Przykład wywołania (formularz płatności):

```
$result = $client->doTransactionBackground([
    'gatewayUrl' => 'https://pay-accept.bm.pl',
    'transaction' => [
        'orderID' => '12345',
        'amount' => '5.12',
        'description' => 'Test transaction 12345',
        'gatewayID' => '1500',
        'currency' => 'PLN',
        'customerEmail' => 'test@test.test',
        'customerIP' => '127.0.0.1',
        'title' => 'Test',
        'validityTime' => date('Y-m-d H:i:s', strtotime('now +5 hour')),
        'linkValidityTime' => date('Y-m-d H:i:s', strtotime('now +5 hour'))
    ]
]);

$transactionBackground = $result->getData();
```

```
echo $transactionBackground; // <form
action="https://pg-accept.blue.pl/gateway/test/index.jsp" name="formGoPBL"
method="POST"><input type="hidden" name="transaction" value="758519"> (...)
```

Obsługa ITN (Instant Transaction Notification)

Serwis BlueMedia po wykonaniu płatności wysyła na wcześniej skonfigurowany adres ITN komunikat o statusie płatności. Dane przesyłane są w formacie XML dodatkowo zakodowanym w base64. SDK oferuje metodę `doItnIn` która w wyniku przekazania danych z serwisu BlueMedia zwraca gotowy obiekt `BlueMedia\Itn\ValueObject\ItnIn` pozwalający na użycie akcesorów lub konwersję do tablicy. Dzięki temu obiektowi, programista może użyć danych potrzebnych np. do aktualizacji statusu płatności w bazie danych itp.

Po przetworzeniu komunikatu ITN należy przekazać odpowiedź. Służy do tego metoda `doItnInResponse` która przyjmuje obiekt `ItnIn` oraz argument informujący o potwierdzeniu transakcji.

Poniżej przykład zastosowania obsługi ITN:

```
$result = $client->doItnIn($_POST['transactions']);

$itnIn = $result->getData();
$transactionConfirmed = $client->checkHash($itnIn);

// Jeżeli status płatności z ITN jest potwierdzony i hash jest poprawny - zakończ
// płatność w systemie
if ($itnIn->getPaymentStatus() === 'SUCCESS' && $transactionConfirmed) {
    $order = $this->orderRepository->find($itnIn->getOrderId());

    $order->setPaymentCompleted();
}

$itnResponse = $client->doItnInResponse($itnIn, $transactionConfirmed);

return new Response($itnResponse->getData()->toXml());
```

Obsługa ITN, utworzenie obiektu komunikatu

Podczas implementacji może okazać się że przed wykonaniem obsługi ITN zajdzie potrzeba np. konfiguracji klienta na podstawie danych dostępowych w oparciu o walutę. W takim modelu programista może wspomóc się metodą `getItnObject`.

```
$itn = Client::getItnObject($_POST['transactions']);

$itn->getCurrency(); // PLN
// ...
```

Pobieranie listy aktualnie dostępnych regulaminów

Metoda `getRegulationList` umożliwia odpytanie o aktualną listę regulaminów wraz linkami do wyświetlenia w serwisie oraz akceptacji przez klienta.

```
$result = $this->client->getRegulationList('https://pay-accept.bm.pl');  
return $result->getData();
```

Pobieranie listy kanałów płatności

Metoda `testGetPaywayList` umożliwia odpytanie o aktualną listę płatności.

```
$result = $this->client->getPaywayList('https://pay-accept.bm.pl');  
return $result->getData();
```