

PRZELEWY ONLINE

DOKUMENTACJA TECHNICZNA

WERSJA	DATA	ZMIANY
1.0	06-06-2018	Udostępnienie oficjalnej wersji API
1.1	18-02-2020	Aktualizacja API
1.2	19-04-2020	Możliwość anulowania transakcji
1.3	21-04-2020	Działanie white-label BLIK

1. GENEROWANIE TRANSAKCJI

Generowanie transakcji opiera się na wysłaniu zapytania przez serwis partnera do API a następnie uzyskania linku na którego należy przekierować klienta.

Wejściowy adres na który należy wysłać zbiór danych do wygenerowania transakcji:

<https://secure.paybylink.pl/api/v1/transfer/generate>

Dane należy przesłać zakodowane w formacie json. Wszystkie stringi enkodowane są w utf-8. Zapytanie należy wysłać na podany adres wejściowy metodą POST. W nagłówku pole 'ContentType' powinno być ustawione na „application/json”

OPIS PRZYKŁADOWEGO ZAPYTANIA

NAZWA	TYP	ROZMIAR	WYMAGANE	OPIS
shopId	int	32bity	TAK	Id sklepu (do odczytania w panelu paybylink)
price	float	32bity	TAK	Kwota jaką klient zostanie obciążony
control	string	0-255 znaki	NIE	Dodatkowe dane, które klient może dodać do zapytania(np id transakcji w systemie klienta, numer zamówienia)
description	string	0-255 znaki	NIE	Opis transakcji
email	string	0-100 znaków	NIE	Jeśli zostanie zawarty w zapytaniu, będzie domyślnie wpisany w polu "Email" w każdej metodzie transakcji.
notifyURL	string	0-300 znaków	NIE	Definiuje adres na który będą wysyłane powiadomienia o dokonanych transakcjach.
returnUrlSuccess	string	0-300 znaków	NIE	Definiuje adres na który klient zostanie przekierowany po udanej transakcji
returnUrlSuccessT idPass	bool	-	NIE	Jeśli ustawiony na true, przekazuje id transakcji w parametrze 'tid' w linku powrotnym po udanej transakcji(returnU rISuccess)

NAZWA	TYP	ROZMIAR	WYMAGANE	OPIS
hideReceiver	bool	-	NIE	Ukrywa nazwe odbiorcy
customFinishNote	string	0-255 znaki	NIE	Umożliwia wyświetlanie swojej informacji w polu dokonania płatności
signature	string	-	TAK	Sygnatura transakcji. Generowanie opisane poniżej.

1. SPOSÓB OBLICZANA SYGNATURY:

Sygnature należy obliczyć poprzez wygenerowania skrótu funkcją wybraną w panelu (z reguły sha256) z połączonych ze sobą: klucza prywatnego i podanych niepustych pól.

Pomiędzy każdą łączoną wartością należy zastosować separator „|”
Kolejność połączonych wartości musi być zgodna z kolejnością pól zawartych w dokumentacji. Pierwszym argumentem jest klucz prywatny zawarty w konfiguracji sklepu w panelu.

Pole 'price' należy zapisać w formacie %.2f

Przykład:

```
sha256hex("SECRET_KEY|241|29.70|id:58081|Przykładowy opis|user@example.com|  
https://example.com/payments/handlers/paybylink/notifications|https://example.com/thank-you")
```

Przykładowe wygenerowane zapytanie przeznaczone do wysłania do API:

```
{  
  "shopId":241,  
  "price":29.7,  
  "control":"tid:58081",  
  "description":"Przykładowy opis",  
  "email":"user@example.com",  
  "notifyURL":"https://example.com/payments/handlers/paybylink/  
notifications",  
  "returnUrlSuccess":"https://example.com/thank-you",  
  "signature":"fa70924e8b249a3d75dba5004675fcb2613a8a3993b  
5e3f30fd674355541cefe"  
}
```

Po wysłaniu tego zapytania, w tej samej sesji http należy odebrać odpowiedź. Weryfikując kod błędu należy sprawdzić czy wygenerowanie transakcji przebiegło pomyślnie.

KOD BŁĘDU	OPIS
400	Błąd w dekodowaniu danych do formatu JSON
401	Wymanage pola nie zostały podane
402	Rozmiar pól nieprawidłowy
403	Nieprawidłowa sygnatura zapytania
404	Id sklepu nie zostało odnalezione
500	Błąd wewnętrzny podczas generowania transakcji
200	Wszystko ok. Transakcja została wygenerowana.

Przykładowa błędna odpowiedź:

```
{  
  "errorCode":404,  
  "error":"shop doesn't exist"  
}
```

Przykładowa prawidłowa odpowiedź:

```
{  
  "url":"https://secure.pbl.pl/transfer/w3WUQXsMgSYc9gH3V8Hcv  
HxazU5TvJLJ",  
  "transactionId":"w3WUQXsMgSYc9gH3V8HcvHxazU5TvJLJ"  
}
```

2. Odbieranie notyfikacji o transakcjach

Jeśli parametr notifyURL został podany, będą na niego dostarczane informacje o płatnościach. Notyfikacje wysyłane są na podany URL metodą POST w body, gdzie przesyłane są następujące dane w formacie JSON:

POLE	TYP	OPIS
transactionId	string – 32 znaki	Unikalne ID transakcji
control	string	dane kontrolne podane przez klienta
email	string	Email podany przez kupującego
amountPaid	float(32 bity)	Kwota wpłacona przez kupującego
notificationAttempt	int	Numer próby dostarczenia notyfikacji
paymentType	string	Występują: „PAYPAL”, „TRANSFER”
apiVersion	int	Wersja API
signature	int	Sygnatura transakcji. KLIENT ZOBOWIĄZANY JEST JĄ SPRAWDZAĆ. Sposób generowania sygnatury jest opisany poniżej.

W języku PHP dane można odebrać za pomocą

```
$response = file_get_contents('php://input');  
$response = json_decode($response, true);
```


Sposób obliczana sygnatury:

Obliczenie sygnatury następuje przez wygenerowania skrótu funkcją wybraną w panelu (z reguły sha256) z połączenie klucza prywatnego i wszystkich pól separatorem „|”.

Wartość pola „amountPaid” musi być zapisana w formacie %.2f(np. 2.00, 40.56, 64.40, 89.01)

Przykład

```
{  
  "transactionId":"w3WUQXsMgSYc9gH3V8HcvHxazU5TvJLJ",  
  "control":"tid:58081", "email":"user@example.com",  
  "amountPaid":29.70, "notificationAttempt":5,  
  "paymentType":"TRANSFER", "apiVersion":1,  
  "signature":"029c7c4f327ed03e5d716b6142a079c855afcc73656  
  eb98e864a27a241b75e9d"  
}
```

Każdą otrzymaną notyfikacje należy potwierdzić odpowiadając w tej samej sesji kodem HTTP 200 OK. W body należy zamieścić „OK”. Nagłówek ‘Content-Type’ powinien być ustawiony na „text/plain”.

W przypadku błędu połączenia lub innym kodem odpowiedzi niż 200 OK lub ciałem odpowiedzi innym niż „OK” zostanie ponowiona próba wysłania powiadomienia.

W przypadku wielokrotnego niepowodzenia, powiadomienia będą wysyłane są z coraz większym odstępem czasowym (zwiększanym wykładniczo)

Maksymalna ilość wysłanych powiadomień, po których transakcja zostanie zakończona z niepowodzeniem to 241.

ANULOWANIE TRANSAKCJI

Link wejściowy

```
https://secure.paybylink.pl/api/v1/transfer/cancel
```

Przykładowy JSON do anulowania transakcji

```
{  
  "shopId":923,  
  "transactionId":"IMoI55QsLRMzoWxDgliR5LpSTu9Oa7iX",  
  "customReason":"customReason test",  
  "signature":"a611c31c1aee000ef1d0d89f2579433a78aa2887f69  
f14a211f4530c1a43c97c",  
}
```

signature generujemy analogicznie do generowania transakcji

Przykładowa odpowiedź

```
{  
  "cancelled":false,  
  "cancelError":"Transaction already cancelled"  
}
```

1. GENEROWANIE TRANSAKCJI BLIK WHITELABEL

Generowanie transakcji opiera się na wysłaniu zapytania przez serwis partnera do API

<https://secure.paybylink.pl/api/v1/transfer/blikauth>

Dane należy przesłać zakodowane w formacie json. Wszystkie stringi enkodowane są w utf-8. Zapytanie należy wysłać na podany adres wejściowy metodą POST. W nagłówku pole 'ContentType' powinno być ustawione na „application/json”

OPIS PRZYKŁADOWEGO ZAPYTANIA

NAZWA	TYP	ROZMIAR	WYMAGANE	OPIS
shopId	int	32bity	TAK	Id sklepu (do odczytania w panelu paybylink)
price	Decimal	32bity	TAK	Kwota jaką klient zostanie obciążony
code	string	0-255 znaki	TAK	6 cyfrowy kod udostępniany w aplikacji bankowej
customerIP	string	-	TAK	adres IP klienta, który generuje transakcje
control	string	0-255 znaki	NIE	Dodatkowe dane, które klient może dodać do zapytania(np id transakcji w systemie klienta, numer zamówienia)
notifyPaymentURL	string	0-300 znaków	NIE	Definiuje adres na który będą wysyłane powiadomienia o dokonanych transakcjach.
notifyStatusURL	string	0-300 znaków	NIE	Definiuje adres na który będą wysyłane powiadomienia o statusie kodu BLIK.
signature	string	-	TAK	Sygnatura transakcji. Obliczana analogicznie do sygnatury standardowej transakcji.

W odpowiedzi zwracany jest parametr transactionId

PAKIETY PRZESŁANE POD ADRES NOTIFYSTATUSURL

NAZWA	TYP	ROZMIAR	WYMAGANE	OPIS
transactionId	string	-	TAK	ID transakcji
control	string	-	TAK	Identyfikator transakcji stworzony przez Sprzedającego
price	Decimal	-	TAK	Kwota transakcji
status	string	-	TAK	Przykładowe statusy poniżej
signature	string	-	TAK	Sygnatura transakcji. Obliczana analogicznie do sygnatury standardowej transakcji.

```
Incomed: {  
  "transactionId": "Wrd2YBJEUDnnFDdMAa17G2jMxn7wVv32",  
  "control": ""  
  , "price": "1.01"  
  , "status": "WRONG_TICKET",  
  "signature": "b2c390617ef4dde34466b6c043e19109df7748d9bbf  
47042e20863be444b7491"  
}
```

Incomed:

```
{  
  "transactionId": "Xpe9LKAGq1MxIZFMRTpoaScUAFz3d2oF",  
  "control": "",  
  "price": "1.01",  
  "status": "USER_DECLINED",  
  "signature": "8d17a21a862437ac3f0f43f2903d8a74a2865441  
b6bfc71a8f2292fcb6126197"  
}
```

Incomed:

```
{"transactionId": "Fp8DwwNcYuV9t0uJswvK86o0ECu0IU37",  
"control": "",  
"price": "1.01",  
"status": "TIMEOUT",  
"signature": "255e2f74c148c30758bf3260462e059dc0b0475f3c  
7d1549f76485b24304b06f"  
}
```

Incomed: {

```
  "transactionId": "Fp8DwwNcYuV9t0uJswvK86o0ECu0IU37",  
  "control": "",  
  "price": "1.01",  
  "status": "CONFIRMED",  
  "signature": "02b3ba5a1ed6467ba29b60d46d21bac51e4a8315  
2605cfef8ebc936c2e885ade"  
}
```


SDK PHP

<https://paybylink.pl/paybylink-sdk.zip>