

## 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.rushpay.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 ‘Content-Type’ powinno być ustawione na „application/json”

Opis przykładowego zapytania:

Nazwa	Type	Rozmiar	Wymanage	Opis
shopId	int	32bity	TAK	Id sklepu(do odczytania w panelu rushpay)
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 znaku	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.
returnUrlSuccessTidPass	bool	-	NIE	Jeśli ustawiony na true, przekazuje id transakcji w parametrze 'tid' w linku powrotnym po udanej transakcji(returnUrlSuccess)
hideReceiver	bool	-	NIE	Ukrywa nazwe odbiorcy
customFinishNote	string	0-255 znaków	NIE	Umożliwia wyświetlanie swojej informacji w polu dokonania płatności
signature	string	Zależy od wybranej metody obliczania skrótu		Sygnatura transakcji. Generowanie opisane poniżej.

### 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. Pomiedzy 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/rushpay/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/rushpay/notifications",
  "returnUrlSuccess":"https://example.com/thank-you",
  "signature":"fa70924e8b249a3d75dba5004675fcb2613a8a3993b5e3f30fd674355541cefe"
}
```

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

#### Kody błędów HTTP:

Kod	Znaczenie
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.rushpay.pl/transfer/w3WUQXsMgSYc9gH3V8HcvHxazU5TvJLJ",  
  "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, 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	string	Sygnatura transakcji. <b>KLIENT ZOBOWIĄZANY JEST JĄ SPRAWDZAĆ.</b> Sposób generowania sygnatury jest opisany poniżej.

### 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:

```
sha256hex("SECRET_KEY|w3WUQXsMgSYc9gH3V8HcvHxazU5TvJLJ|tid:58081|user@example.com|654.30|5|TRANSFER|1")
```

Przykładowe ciało odebranej notyfikacji:

```
{
  "transactionId": "w3WUQXsMgSYc9gH3V8HcvHxazU5TvJLJ",
  "control": "tid:58081",
  "email": "user@example.com",
  "amountPaid": 29.70,
  "notificationAttempt": 5,
  "paymentType": "TRANSFER",
  "apiVersion": 1,
  "signature": "029c7c4f327ed03e5d716b6142a079c855afcc73656eb98e864a27a241b75e9d"
}
```

Każdą otrzymaną notyfikację 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.