Payment Gateway technical information

Structure

The webshops need to be connected solely to BIG FISH transaction server that will provide all the communication tasks related to the realization of transactions with the payment service providers chosen by the webshop.

Payment Gateway SDK:

Payment Gateway SDK makes the integration procedure of payment services to the webshops or applications faster and easier. SDKs below contain code examples, equation libraries and documentation what is necessary to reach the functions of Payment Gateway.

PHP SDK: https://www.paymentgateway.hu/static/downloads/sdk/php.zip
iOS SDK: https://www.paymentgateway.hu/static/downloads/sdk/ios.zip
Android SDK: https://www.paymentgateway.hu/static/downloads/sdk/android.zip

There are two important elements in the configuration file of the SDKs above:
[BIGFISH_PAYMENTGATEWAY_STORE_NAME_] (shop identification)
[BIGFISH_PAYMENTGATEWAY_API_KEY] (API key).

To request Shop ID and API key please contact us!

Payment Gateway API:

If you would like to use Payment Gateway with a different technology, you can use APIs bellow:

SOAP API

-   test system: https://test.paymentgateway.hu/api/soap/?wsdl
-   live system: https://www.paymentgateway.hu/api/soap/?wsdl

REST API

-   test system: https://test.paymentgateway.hu/api/rest/
-   live system: https://www.paymentgateway.hu/api/rest/

Authentication

HTTP basic access authentication: http://en.wikipedia.org/wiki/Basic_access_authentication

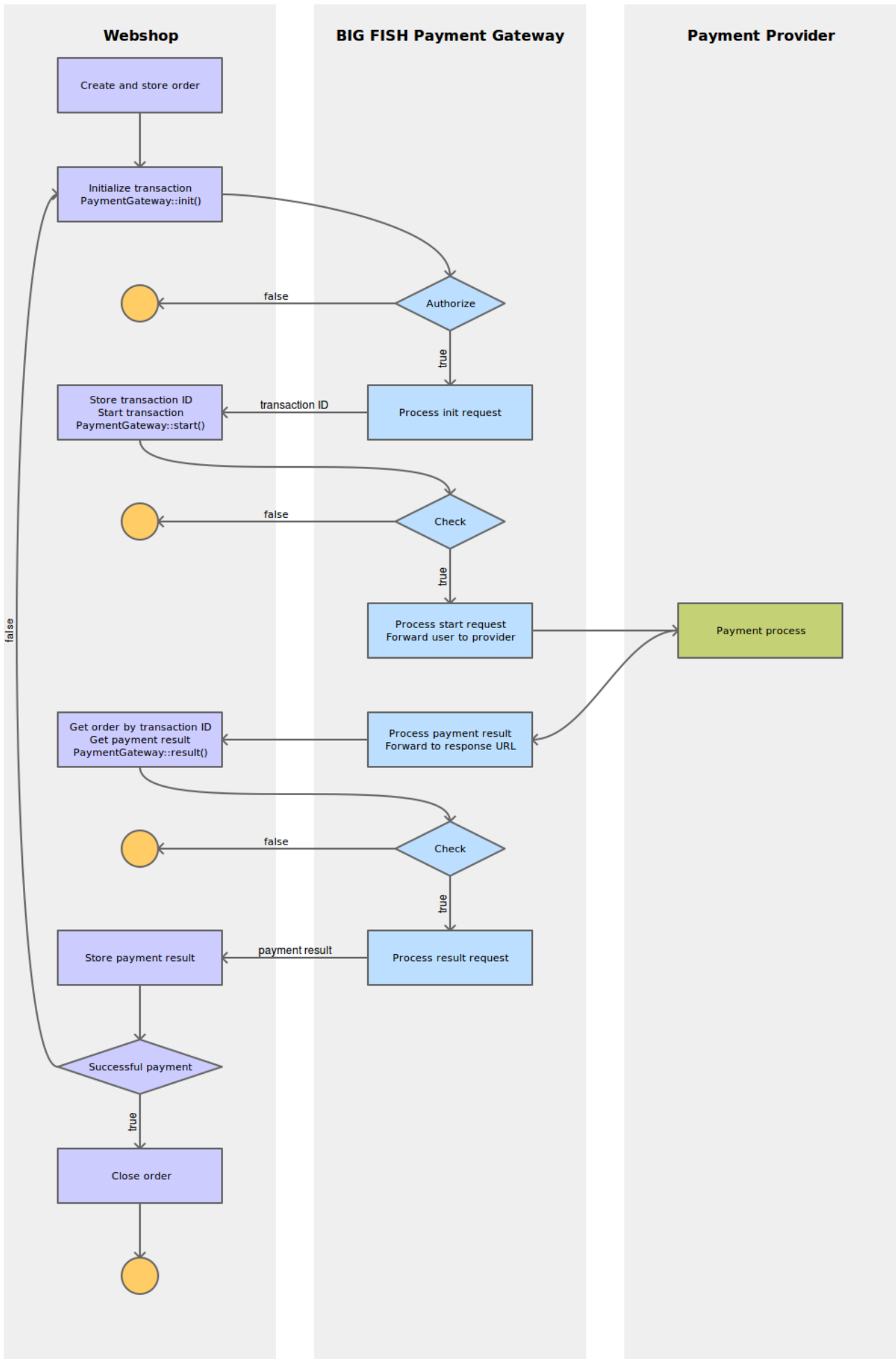Usage:

Please send it in HTTP REQUEST header,

   Authorization: Basic Ym9sdG5ldmU6NGQ4NDgtYzhhYzAtMmJkZTctNWE3YzktNDcxZGY=

where in case of *Ym9sdG5ldmU6NGQ4NDgtYzhhYzAtMmJkZTctNWE3YzktNDcxZGY=* you can create it this way:

   base64_encode("store_name:api_key")

To request Shop ID and API key please contact us!

The procedure of payment transaction

Webshop	BIG FISH Payment Gateway	Payment Provider

Create and store order

Initialize transaction
PaymentGateway::init()

Authorize — false

true

Process init request

Store transaction ID
Start transaction
PaymentGateway::start()

transaction ID

Check — false

true

Process start request
Forward user to provider

Payment process

Get order by transaction ID
Get payment result
PaymentGateway::result()

Process payment result
Forward to response URL

Check — false

true

Store payment result

payment result

Process result request

Successful payment

false

true

Close order

1. Customer finalizes the order and chooses electronic payment.
2. Webshop registers the order in its own database in pending status.
3. Webshop sends a message to Payment Gateway in the background about the new transaction (Init).
4. Payment Gateway confirms the initialization of the successful transaction to Webshop.
5. Webshop saves the transaction ID received from Payment Gateway and sends Customer to Payment Gateway to pay (Start).
6. Payment Gateway realizes the payment between the payment service provider and the Customer.
7. Payment Gateway confirm the result of payment to Customer.
8. Payment Gateway sends the Customer back to Webshop page.
9. Webshop requests the result of payment from Payment Gateway (Result).
10. Webshop – depending on the result of payment – in case of successful payment finalize the order, or in case of failed payment offers the possibility of retrying or offers an alternative payment method to the Customer.

Payment Gateway functions

1. INIT

Init function is for initialisation of a new payment transaction. The communication between Webshop and Payment Gateway is in the background, hidden from Customer.

1.1. Example for initialisation of transaction generated from Webshop (PHP SDK)

```
$request = new PaymentGateway_Init_Request();
$request->setProviderName("Sofort")
    ->setResponseUrl("http://www.webshop.com/payment_gateway_response_url.php")
    ->setAmount(10)
    ->setCurrency("EUR")
    ->setOrderId("order123")
    ->setUserId("user123")
    ->setLanguage("EN");
$response = PaymentGateway::init($request);
```

1.2. REST call (HTTP POST)

method=Init&json={"StoreName":"storename","ProviderName":"Sofort","ResponseUrl":"http:\/\/www.webshop.com\/payment_gateway_response_url.php","Amount":"10","Currency":"EUR","OrderId":"order123","UserId":"user123","Language":"EN"}

1.3. Explanation of parameters

Parameter	Mandatory?	Default value	Type	Value set
ProviderName	Mandatory	No	Text (max. 20 characters)	o CIB o Escalion o KHB (K&H Bank) o MPP (Mobile Payment) o MPP2 (MasterCard Mobile) o OTP o OTP2 (two participants) o PayPal o PayU o SMS o QPAY (Wirecard QPAY) o Unicredit (UniCredit Bank)

				o FHB (FHB Bank)
				o Barion
				o PayUMobile (PayU Mobile)
				o PayUWire (PayU Wire)
				o PayUCash (PayU Cash)
				o OTPay (OTPay Mobile Payment)
				o Sofort (Sofort Banking)

This parameter determines the chosen payment service provider.

StoreName	Mandatory	No	Text (max. 20 characters)	Defined when contract is signed.

Unique ID of Webshop in Payment Gateway.

ResponseUrl	Optional or Mandatory	Depends on settings	Text (max. 225 characters)	Optional

URL address, where Payment Gateway sends Customer after the transaction. It is possible to setup the default response address on Payment Gateway administration interface, in case it is setup, there is no need to fill it always when we iniciate a new transaction.

Amount	Mandatory	No	Positive real number	Optional

Gross total amount that Customer has to pay.

OrderId	Optional	No	Text (max. 225 characters)	Optional

The order number at the Webshop, it is strongly proposed to help to find the transactions according to different criterias.

UserId	Optional	No	Text (max. 225 characters)	Optional

The Customer's ID at the Webshop, it is strongly proposed to help to find the transactions according to different criterias.

Currency	Optional	HUF	Text (3 characters)	o HUF o EUR o USD

Currency (can be different / payment service providers)

Language	Optional	HU	Text (2 characters)	o HU o EN o DE

This determines the language of the site.

MppPhoneNumber	Optional	No	Text (max. 32 characters)	Optional (for example: 36201114567)

At Mobile Payment the cellphone number of the user can be provided in advance.

AutoCommit	Optional	true	Text	o true o false

It will order the bank to realize the transaction immediately or only block it first and we will confirm it later or cancel it with the CLOSE function.

1.4. Response from Payment Gateway

Parameter	Type	Value set	Example
TransactionId	Text (32 character long md5 hash)	Changing	0a62408585bdfbb54851fd454078641d

Transaction ID generated by Payment Gateway in case of successful transaction.

Parameter	Type	Value set	Example
ResultCode	Text	SUCCESSFUL and so on	UnknownProvider

Result code shows the success of the initialisation of the transaction. If the result code is different than „SUCCESSFUL", initialisation was unsuccessful.

Parameter	Type	Value set	Example
ResultMessage	Text	Changing	Unknown Payment Provider

Text explanation regarding the result code above (in the language chosen at the initialisation of the transaction). If the Webshop requests, it can be shown to the Customer.

2. START

Webshop can send the Customer to Payment Gateway with START function to realize the payment. (HTTP Redirect)

2.1 Example for Start function call (PHP SDK)

PaymentGateway::start(new PaymentGateway_Start_Request($response->TransactionId));

2.2 Example for URL call (HTTP Redirect)

https://test.paymentgateway.hu/Start?TransactionId=0a62408585bdfbb54851fd454078641d

2.3 Explanation of parameters

Parameter	Mandatory?	Default value	Type	Value set
TransactionId	Mandatory	No	Text (32 characters long md5 hash)	Changing

Transaction ID received at initialisation.

2.4 Response received from Payment Gateway

After realizing the transaction, Payment Gateway sends Customer to the response URL received from Webshop, for example:

http://www.webshop.com/payment_gateway_response_url.php?TransactionId=0a62408585bdfbb54851fd4540 78641d

3. RESULT

Webshop can ask for the result of the transaction from Payment Gateway with Result function.
The communication between Webshop and Payment Gateway is in the background, hidden from Customer.

3.1 Example for Result function call (PHP SDK)

$response = PaymentGateway::result(new PaymentGateway_Result_Request($_GET["TransactionId"]));

3.2 REST call (HTTP POST)

method=Result&json={"TransactionId":"0a62408585bdfbb54851fd454078641d"}

3.3 Explanation of parameters

Parameter	Mandatory?	Default value	Type	Value set
TransactionId	Mandatory	No	Text (32 characters long md5 hash)	Changing

Transaction ID received in response URL.

3.4 Response received from Payment Gateway

Parameter	Type	Value set	Example
TransactionId	Text (32 character long md5 hash)	Changing	0a62408585bdfbb54851fd454078641d

Transaction ID registered in Payment Gateway.

Parameter	Type	Value set	Example
ResultCode	Text	o SUCCESSFUL o PENDING o ERROR o CANCELED	SUCCESSFUL

Result code shows the success of the transaction.

Parameter	Type	Value set	Example
ResultMessage	Text	Changing	Successful transaction

Text explanation regarding result code above (in the language chosen at the initialisation of the transaction). If the Webshop requests, it can be shown to the Customer.

Parameter	Type	Value set	Example
Anum	Text (max. 100 characters)	Changing	AN123456

Authorisation code or authorisation number received from payment service provider (optional).

Parameter	Type	Value set	Example
OrderId	Text (max. 255 characters)	Changing	12345

The number of the order at Webshop. It helps to find the order later.

Parameter	Type	Value set	Example
UserId	Text (max. 255 characters)	Changing	123

Customer's ID at Webshop. It helps to find the order later.

Parameter	Type	Value set	Example
ProviderTransactionId	Text (max. 255 characters)	Changing	5467429862223832

Payment service provider specific transaction ID. It helps to find the order later.

Parameter	Type	Value set	Example
AutoCommit	Text	o true o false	true

If the transaction was completed financially immediately, the value is **true.** In case it shall be confirmed later – with Close function – the value is **false.**

CommitState	Text	o PENDING o APPROVED o DECLINED	APPROVED

In case we confirmed the debit with Close function, it is APPROVED, if we refused, it is DECLINED.

4. CLOSE

It is possible to finalize the pending transactions with Close function by resolving the blocked amount or debiting the Customer's bank account.

4.1 Example for Close function call (PHP SDK)

$response = PaymentGateway::close(new PaymentGateway_Close_Request($transactionId, true));

4.2 REST call (HTTP POST)

method=Close&json={"TransactionId":"0a62408585bdfbb54851fd454078641d","Approved":"true"}

4.3 Explanation of parameters

Parameter	Mandatory?	Default value	Type	Value set
TransactionId	Mandatory	No	Text (32 characters long md5 hash)	Changing

Transaction ID registered in Payment Gateway.

Approved	Optional	true	Text	o true o false

True debits Customer's account, **false** resolve the blocked amount.

4.4 Response received from Payment Gateway

Parameter	Type	Value set	Example
TransactionId	Text (32 characters long md5 hash)	Changing	0a62408585bdfbb54851fd454078641d

Transaction ID registered in Payment Gateway.

ResultCode	Text	o SUCCESSFUL o ERROR	SUCCESSFUL

Result code shows success of realization.