Dotpay Technical Support
Wielicka Str. 72, 30-552 Cracow, Poland
phone. +48 12 688 26 00
e-mail: tech@dotpay.pl

Worldwide secure payment

TECHNICAL MANUAL FOR PAYMENTS IMPLEMENTATION

version 1.29.11.1

TABLE OF CONTENT

INTRODUCTION

This document describes the payment system interface offered by Dotpay service. It is designed both for those interested in offering the service, as well as those involved in the implementation of the system.

I. RECEIVING PAYMENTS

There are three methods of accepting payments from customers. The first two methods (described in subchapters *Basic version* and *Generated version*) are intended for Sellers who do not have e-shops or services integrated with the Dotpay system. The third method (presented in subchapter *Direct version*) is designed for e-shops and services, at which the payment process begins (e.g. customer is ordering some item).

I. 1. Basic version

The customer who makes a payment for the seller who has their own ID number (e.g. `id=123456`) should use **"Pay"** link which is on the main website www.dotpay.pl or directly from https://ssl.dotpay.pl/t2/ .

After typing the link mentioned above into web browser address field, the buyer should enter the following data:

1. Seller's ID (e.g. `123456`)
2. transaction's amount (e.g. `12.42`)
3. transaction's currency (e.g. `PLN`)
4. transaction's description (e.g. `Payment for invoice #12345/2014`)

You can use the following method to enable the buyer change the amount and the description manually (in case of donation).

Any of above-mentioned data can be provided via `URL` address, e.g. to eliminate possibility of entering the wrong sellers ID, you can add it to the redirection address. In this case you should provide the buyer with following `URL` address:

https://ssl.dotpay.pl/t2/?id=123456

In the above link ID of the seller is passed to Dotpay payment form with use of `GET` method. The same method is used to pass the other payment parameters – in next example we will send `amount` and `description` to payment form:

https://ssl.dotpay.pl/t2/?id=123456&amount=123.00&description=Test

If `currency` parameter has NOT been sent, the transaction will be processed in default currency set for seller account ID `123456`.

Other parameters that can be passed in this way to Dotpay payment form are listed in subchapter *Direct version* of this document.

I. 2. Generated version

After login to My Dotpay panel you can use menu *Tools → Payment links generator*. This menu allows to determine such data as: `id`, `amount`, `currency`, `description` and `language`. Generating a payment request means creating a special key which allows defining the given parameters thus payment process optimization.

Keys generated this way can be listed and modified (e.g. `amount`), in the above mentioned menu.

Key example: `rfhu4jb5ym657g3xluf4bbqfmbyj6t17`

Using such a generated key you can elicit payment site from the link:

https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbyj6t17

In case when payment language was NOT defined or defined language was erased, usage of `lang` parameter allows you to determine language on payment page.

Example of payment page with added parameter `lang=en` to present payment form in English:

https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbyj6t17&lang=en

Such a prepared link with a key can be used in on-line auctions, where pasted on auction page, or sent by e-mail to a buyer, will allow a quick payment reception.

Generated payment link with a key can be used repeatedly until erased by the Merchant in My Dotpay panel (*Tools → Payment links generator → delete*).

I. 3. Direct version

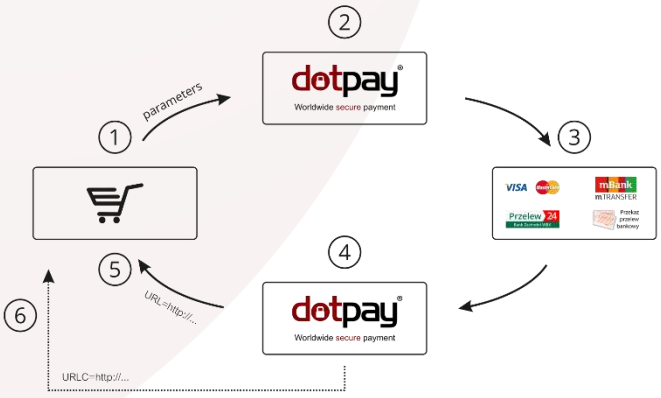
Direct version is based on direct integration with Dotpay payment system. In this version a customer after making an order will be redirected from the seller's web site to Dotpay payment form. Client located on the Seller's web page, after clicking on the button to confirm payment (e.g. **"pay by Dotpay"**) should be redirected to https://ssl.dotpay.pl/t2/ .

In the next step the customer makes the payment with use of selected payment channel. If there is a defined return address, on the payment confirmation page a button to return to the seller web page will be displayed. Return address and return button are defined by giving appropriate values to parameters described below `URL` and `type`.

A seller may also specify `URLC` address on which information about the status of the transaction will be sent via `POST` method. The address can be defined by logging into Dotpay or sent when you initiate the payment as a parameter named `URLC`. It should be defined in the My Dotpay panel (*Settings → Notifications → URLC Configuration → Edit*) option: **"Block external URLC"**. With such a configuration a seller can define other `URLC` addresses for each payment.

Data sent to the `URLC` address as a confirmation of the transaction are described in chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)*.

Example of the payment process diagram:



1. The customer having clicked **"Pay"** (or similar) button is redirected to the Dotpay payment form (https://ssl.dotpay.pl/t2/). Simultaneously with the `POST` method (or `GET`) you can send payment parameters:
    a) Seller's ID (e.g. `id=123456`)

b) transaction's amount (e.g. `amount=12.42`)

c) transaction's currency (e.g. `currency=PLN`)

d) transaction's description (e.g. `Payment for invoice #12345/2014`)

You can also submit other optional parameters. Description of all parameters that can be transferred to a payment form are in below placed tables.

2. In a payment form the customer will choose the channel with which he wants to make a payment (e.g. Bank or Card).
3. The client is redirected to chosen payment channel (logging page of bank or credit card payment form)
4. Payment confirmation will be shown on Dotpay page.
5. If the return address to the seller's web page has been defined, a customer will see a button that redirects to defined address for example:

   `URL=http://www.example.com/thanks_page.php`

6. If the `URLC` address has been defined, confirmation of current transaction status will be sent to this address (regardless of the button to return to the site of the Seller), for example:

   `URLC=http://www.example.com/urlc_receiver.php`

The set of basic parameters sent to the Dotpay payment form are presented in the below table. These parameters should be passed via `POST` (or `GET`) method to the following address https://ssl.dotpay.pl/t2/ .

Table 1. (Basic parameters)

PARAMETER	MEANING / DESCRIPTION
`api_version`	API version - according to which the system sends `URLC` confirmation (see chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)*). Available values: `dev` - API version described in this manual. **WARNING !** Lack of sending the parameter in redirect causes sending by Dotpay system unsupported version of URLC notification which is not accordance with notification described in this instruction.
`id`	Account ID for which payment is made. type: `integer` min value: `1` max value: `999999`
`amount` `kwota`	The amount of the transaction given with centesimal part (two places after the separator). Centesimal separator is a dot (.)

	type: `string` min value: `1` max value: `10` Example: `amount=42.82`
`currency` `waluta`	This parameter determines the currency in which the `amount` parameter is given. Format according to ISO 4217 standard. Available values: `PLN` (default), `EUR`, `USD`, `GBP`, `JPY`, `CZK`, `SEK` Example: `currency=EUR`
`description` `opis`	Payment description. type: `string` min value: `1` max value: `255` Example: `description=Invoice 20/2014`
`lang` `jezyk`	Defines the language of web page and payment forms. Available values: `pl` – polish (default), `en` – english, `de` – german, `it` – italian, `fr` – french, `es` – spanish, `cz` – czech, `ru` – russian, `bg` – bulgarian Example: `lang=en`

Below there is a list of additional parameters that if sent together with the basic parameters (above table) will increase the functionality of the payment system.

Table 2. (Additional parameters)

PARAMETER	MEANING / DESCRIPTION
`channel` `kanal`	This parameter defines which payment channel will be selected after redirecting client to Dotpay payment form. Numerical values only available (see *ATTACHMENT 1 - PAYMENT CHANNELS*).

	Example: `channel=3` Seller wishes to select a MultiTransfer channel by default..
`ch_lock` `blokuj`	Forcing the channel specified in the `channel` parameter. The parameter `ch_lock` with a value `1` does not allow the Buyer to choose another channel of payment than defined on the Seller web pages. Available values: `1` – force the selected channel. `0` – channel will not be forced (default) Example: Sending parameters `channel=3` and `ch_lock=1` will not allow a customer to choose other payment method than MultiTransfer.

`channel_groups`
`grupykanalow`
	This parameter defines which channel groups will be displayed after redirecting client to Dotpay payment form.
	
	Available values:
	`K` – payment cards,
	`T` – fast transfers,
	`P` – transfers,
	`G` – cash,
	`W` – purses and vouchers,
	`R` – installments,
	`I` – other,
	`O` – postponed payments.
	
	Parameter can have more than one group as a value, in this case letters have to be separated with commas.
	
	Example:
	`channel_groups=T`
	`channel_groups=R,O`
	`channel_groups=K,T,P`

`URL`	This parameter defines the Internet address (`HTTP` or `HTTPS`) to which Buyer is expected to return after payment. Behavior of `URL` parameter is determined by `type` parameter. type: `string` max length: `1000` Example: `URL=http://www.example.com/thanks_page.php` When both `URL` and `type` parameters are sent (`type=0` and `type=3`) after completion of the payment process customer will see the return button to the

	site of the Seller. The use of the button will move the buyer to the address in the URL parameter with the parameter status (submitted by the POST and GET), which contains information about the possible errors at the checkout pages (OK or FAIL). **WARNING !** Parameter status informs only about the payment process. Information about the current status of the transaction (new, completed, rejected, etc.) contains a variable operation_status sent to the address URLC. Example: URL=http://www.example.com/thanks_page.php?status=OK
type typ	This parameter defines a method of redirection to the seller's web page. Parameter value influences URL parameter. Available values: 0 – after the payment, the buyer will see a button to return to the seller's web page. 1 – after the payment, unseen connection will be established to the address defined in the URL parameter and data sent by POST, chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)*. 2 – no reaction, nothing is sent, lack of a button (default value). 3 – both actions will be made: actions as for parameters type=0 and type=1 – data described in chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)* will be sent via unseen connection (POST method) and return button will be displayed. If URLC notification mechanism is used, the use of this value is redundant. 4 – a direct redirection to the payment channel provider (e.g. bank) as well as after payment making and logging out of the service. This functionality is available only after sending a set of payment parameters required for payments by a given channel and a special configuration on Dotpay site. Example: type=0 URL=http://www.example.com/thanks_page.php After sending the given parameters set a button will be displayed allowing a Buyer to redirect to http://www.example.com/thanks_page.php type=3 URL=http://www.example.com/thanks_page.php After sending the given parameters set a button will be displayed to a Buyer allowing to go back to the website: http://www.example.com/thanks_page.php Data presented in chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)* will be sent via unseen connection (POST method). type=4

	`URL=http://www.example.com/thanks_page.php` After sending the given parameters set and a full set of parameters required for payment process a buyer will be redirected directly (omitting Dotpay interface) to the payment channel provider (e.g. internet banking login form). After logging out of the payment provider service a buyer will be redirected to `URL` address. Using the given value allows a scheme: *Shop-> Bank-> Shop*.
`buttontext` `txtguzik`	The content that appears on a button to return to the seller website. type: `string` min length: `4` max length: `100` Example: `buttontext= Back to www.example.com`
`URLC`	Web address to receive parameters confirming or denying completion of the transaction. A detailed description of `URLC` confirmation is described further in chapter *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)*. **WARNING !** If option **"Block external urlc"** (menu *Settings → Notifications → Urlc configuration → Edit*) is checked, `URLC` parameter will be ignored. type: `string` max length: `1000` Example: `URLC=http://www.example.com/urlc_receiver.php`
`control`	Control parameter allows you to store sequence of data. Unchanged parameter is sent to the Merchant's service in `URLC` confirmation. type: `string` max length: `1000` Example: `control=ec4bf09d3dbe0cb71e6abc3ea44a7273`

`firstname`
`forename`
`imie`
	This parameter identifies the first name of the person making the payment.
	
	type: `string`
	max length: `50`
	
	Example: `firstname= John`

`lastname`
`surname`
`nazwisko`
	This parameter identifies the surname of the person making the payment.
	
	type: `string`
	max length: `50`
	
	Example: `lastname= Smith`

`email`	E-mail address of a person making the payment. Payment confirmation will be sent at this address.

	type: string max length: 100 Example: email= johnsmith@example.com
street ulica	Street name. A part of the address of the buyer. type: string max length: 100 Example: street=Wielicka
street_n1 budynek	Building number. A part of the address of the buyer. type: string max length: 30 Example: street_n1=4
street_n2 mieszkanie lokal	Flat number. A part of the address of the buyer. type: string max length: 30 Example: street_n2=18
state addr2	Additional address data. A part of the address of the buyer. type: string max length: 50 Example: addr2=NY
addr3	Additional address data. A part of the address of the buyer. type: string max length: 50
city miasto	City name. A part of the address of the buyer. type: string max length: 50 Example: city=Kraków
postcode kod	Postal code. A part of the address of the buyer. type: string max length: 20 Example: postcode=30-552
phone	Buyer's phone number.

telefon	type: string max length: 20 Example: phone=127654321
country kraj	Country name. A part of the address of the buyer. The name should be given in accordance with the ISO 3166-1 format (alfa-2 or alfa-3). type: string max length: 50 Example: country=PL country= POL country= Polska
p_info	With use of this parameter a seller can define the name of a payment receiver which will be displayed on the payment Dotpay website. In case the name is not provided the default name will be displayed. type: string max length: 300 Example: p_info=Shop www.example.com
p_email	This parameter describes an e-mail address, which will be seen by a customer. It overwrites an email address provided during registration process. type: string max length: 100 Example: p_email=office@example.com
blik_code	BLIK code confirming payment with given channel. In standard procedure the code is entered by customer on payment provider website after choosing BLIK channel. Sending the parameter in redirection allows shortening payment process because code is entered on shop's website with other order data. type: string Example: blik_code=264230

II. RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)

In order to inform the Seller about the executed operation (transaction) an URLC confirmation mechanism has been created (HTTP request asynchronous connection, callback) which is sent by POST method and fully independent from a Buyer's action.

Confirmations are directed to the Seller's address which can be set in settings of a given shop in Dotpay panel (menu *Settings → Notifications → Urlc configuration → Edit*) or to the address read from URLC sent by the Seller's system in redirection of a Buyer to the payment site (unless URLC parameter has been unblocked from external sources, i.e. uncheck **"Block external URLC"** option). If URLC notification address don't use HTTP secure protocol (i.e. HTTPS) remember to deactivate **"HTTPS verify"** and **"SSL certificate verify"** options in menu mentioned above.

WARNING !

For correctly received URLC confirmation by the Seller's system Dotpay expects an answer

OK

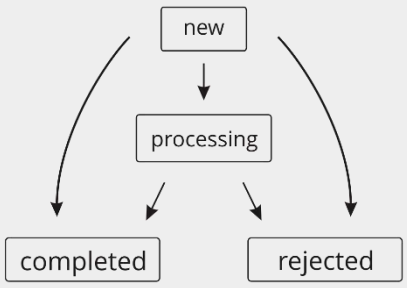
(only two letters, nothing more). **OK** is a correctly received and processed confirmation.

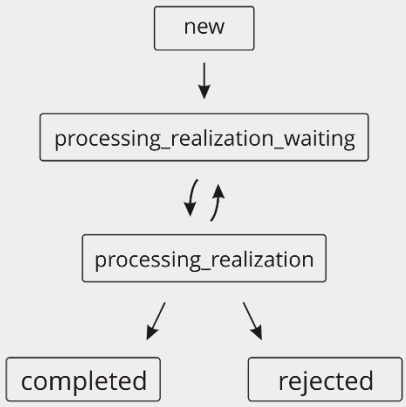
When a different answer than **OK** is reversed by the Seller's system Dotpay system will repeat confirmations (every few, several minutes).

Parameters and optional values sent in URLC confirmations to the Seller's system have been presented in the table below:

Table 3. (Parameters sent by Dotpay back to merchant's system)

PARAMETER	MEANING / DESCRIPTION
id	Account ID type: integer min value: 1 max value: 999999
operation_number	Transaction number. Format according to regular expression: M[0-9]{4,5}-[0-9]{4,5} Example: M1234-5678
operation_type	Transacion type. Available values: payment payment_multimerchant_child payment_multimerchant_parent refund payout

	release_rollback unidentified_payment complaint Example: operation_type= payment
operation_status	Transaction status. Available values: new processing completed rejected processing_realization_waiting processing_realization Example: operation_status=completed Detailed description of operation statuses are presented in chapter *ATTACHMENT II - OPERATION STATUSES DESCRIPTION* **WARNING !** completed or rejected status is final and will not change in the future. Below, you can see a diagram showing statuses of a given payment types (operation_type parameter): for types: payment, payment_multimerchant_child, payment_multimerchant_parent, release_rollback  for types: refund, payout, complaint

	
`operation_amount`	Transaction amount. The decimal separator is a dot (.) type: `string` min length: `1` max length: `10` Example: `operation_amount=177.27`
`operation_currency`	The currency in which the amount parameter is given. Format according to ISO 4217 standard. Example: `operation_currency=PLN`
`operation_withdrawal_amount`	Optional parameter indicating the operation withdrawal amount. Example: `operation_withdrawal_amount=176.00`
`operation_commission_amount`	Optional parameter indicating the operation commission amount. The parameter is presented as a negative amount, and therefore contains a "-" sign. Example: `operation_commission_amount=-1.27`
`operation_original_amount`	The amount taken from `amount` parameter which has been sent by merchant's system. type: `string` min length: `1` max length: `10` Example: `operation_original_amount=42.82`
`operation_original_currency`	The currency taken from `currency` parameter which has been sent by merchant's system. Example: `operation_original_amount=EUR`

operation_datetime	Date and time of transaction's status change. Format: YYYY-MM-DD hh:mm:ss Example: operation_datetime=2014-06-01 12:06:37
operation_related_number	Number of related transaction, if any exists (e.g. in case of a refund). Format according to regular expression: M[0-9]{4,5}-[0-9]{4,5} Example: operation_related_number=M1234-5678 If a confirmation regards refund number M9876-5432 executed for an original operation M1234-5678 the parameter will accept original transaction number value which is M1234-5678.
control	This value corresponds with control parameter which has been sent by merchnat's service in a redirection to Dotpay service while making a payment (see chapter *Direct version*). type: string max length: 1000 Example: control=ec4bf09d3dbe0cb71e6abc3ea44a7273
description	This value corresponds with description parameter which has been sent by merchnat's service in a redirection to Dotpay service while making a payment (see chapter *Direct version*). type: string min length: 1 max length: 255 Example: description= Invoice 20/2014
email	E-mail address of a person making the payment. type: string max length: 100 Example: email=johnsmith@example.com
p_info	Name of a payment receiver which was displayed on the Dotpay payment website during Customer payment making. type: string max length: 300
p_email	E-mail address which was displayed for customer as contact to Seller.

	type: `string` max length: `100`
`channel`	Payment channel (method) used by the buyer. For available values see *ATTACHMENT 1 - PAYMENT CHANNELS*.
`channel_country`	Optional parameter indicating the country of the payment instrument from which payment was made. Format: ISO 3166-1 (alpha-3). Example: `channel_country=POL`
`geoip_country`	Optional parameter indicating the location of the country resulting from IP address from which the payment was made. Format: ISO 3166-1 (alpha-3). Example: `geoip_country=POL`
`signature`	The checksum created by an acronym function `SHA-256` being a concatenation of the above mentioned parameters according to the following formula (if in sent notification some parameter is not present, should be considered as null value or omitted in the calculation): `PIN + id + operation_number + operation_type + operation_status + operation_amount + operation_currency + operation_withdrawal_amount + operation_commission_amount + operation_original_amount + operation_original_currency + operation_datetime + operation_related_number + control + description + email + p_info + p_email + channel + channel_country + geoip_country`  Sign "+" in the above formula has been used only for its readability. It is **not** used to calculate the checksum.  An example of signature parameter calculation in PHP.:  `<?php` `$PIN = "Np3n4QmXxp6MOTrLCVs905fdrGf3QIGm";`  `$sign=` `$PIN.` `$_POST['id'].` `$_POST['operation_number'].` `$_POST['operation_type'].` `$_POST['operation_status'].` `$_POST['operation_amount'].` `$_POST['operation_currency'].` `$_POST['operation_withdrawal_amount'].`

```
        $_POST['operation_commission_amount'].
        $_POST['operation_original_amount'].
        $_POST['operation_original_currency'].
        $_POST['operation_datetime'].
        $_POST['operation_related_number'].
        $_POST['control'].
        $_POST['description'].
        $_POST['email'].
        $_POST['p_info'].
        $_POST['p_email'].
        $_POST['channel'].
        $_POST['channel_country'].
        $_POST['geoip_country'];

    $signature=hash('sha256', $sign);
?>
```

WARNING !
`PIN` used to calculate a `signature` parameter is a string of characters which should be generated by a Merchant in My Dotpay panel (menu *Settings → Notifications → Urlc configuration*).

WARNING !
Lack of verification of a `signature` parameter on a merchant's site is unsecure and may cause financial loss.

III. ADDITIONAL FEATURES

III. 1. Redirection parameters integrity protection (CHK)

In order to protect redirection parameters integrity Dotpay system allows to sign their values. Signature should be sent as additional parameter called `chk` (alongside rest of the payment parameters). `chk` value is checksum calculated with `SHA-256` algorithm using parameter concatenation described below (order must be kept).

```
PIN + api_version + charset + lang + id + amount + currency + description + control +
channel + credit_card_brand + ch_lock + channel_groups + onlinetransfer + url + type
+ buttontext + urlc + firstname + lastname + email + street + street_n1 + street_n2 +
state + addr3 + city + postcode + phone + country + code + p_info + p_email + n_email
+ expiration_date + recipient_account_number + recipient_company +
recipient_first_name + recipient_last_name + recipient_address_street +
recipient_address_building + recipient_address_apartment + recipient_address_postcode
+ recipient_address_city + warranty + bylaw + personal_data + credit_card_number +
credit_card_expiration_date_year + credit_card_expiration_date_month +
credit_card_security_code + credit_card_store + credit_card_store_security_code +
credit_card_customer_id + credit_card_id + blik_code + credit_card_registration +
recurring_frequency + recurring_interval + recurring_start + recurring_count
```

Warning !

Character "+" was used only for visual clarity. It is **NOT** used to calculate the signature.

If given parameter is not used in redirection it has to be skipped.

By default shop (ID) verifies `chk` parameter only if it's sent in payment request (if sum has incorrect value it will result in error, if it's not sent verification won't occur). To force `chk` verification for given ID please send an email to tech@dotpay.pl .

Below is exemplary function (PHP) generating POST/GET request with `chk` .

```php
<?php

  function GenerateChkDotpayRedirection ($DotpayId, $DotpayPin, $Environment,
    $RedirectionMethod, $ParametersArray) {

    $ParametersArray['id'] = $DotpayId;

    $ChkParametersChain =
      $DotpayPin.
       (isset($ParametersArray['api_version']) ?
        $ParametersArray['api_version'] : null).
       (isset($ParametersArray['charset']) ?
        $ParametersArray['charset'] : null).
       (isset($ParametersArray['lang']) ?
        $ParametersArray['lang'] : null).
       (isset($ParametersArray['id']) ?
        $ParametersArray['id'] : null).
       (isset($ParametersArray['amount']) ?
        $ParametersArray['amount'] : null).
       (isset($ParametersArray['currency']) ?
        $ParametersArray['currency'] : null).
       (isset($ParametersArray['description']) ?
        $ParametersArray['description'] : null).
```

```
(isset($ParametersArray['control']) ?
 $ParametersArray['control'] : null).
(isset($ParametersArray['channel']) ?
 $ParametersArray['channel'] : null).
(isset($ParametersArray['credit_card_brand']) ?
 $ParametersArray['credit_card_brand'] : null).
(isset($ParametersArray['ch_lock']) ?
 $ParametersArray['ch_lock'] : null).
(isset($ParametersArray['channel_groups']) ?
 $ParametersArray['channel_groups'] : null).
(isset($ParametersArray['onlinetransfer']) ?
 $ParametersArray['onlinetransfer'] : null).
(isset($ParametersArray['url']) ?
 $ParametersArray['url'] : null).
(isset($ParametersArray['type']) ?
 $ParametersArray['type'] : null).
(isset($ParametersArray['buttontext']) ?
 $ParametersArray['buttontext'] : null).
(isset($ParametersArray['urlc']) ?
 $ParametersArray['urlc'] : null).
(isset($ParametersArray['firstname']) ?
 $ParametersArray['firstname'] : null).
(isset($ParametersArray['lastname']) ?
 $ParametersArray['lastname'] : null).
(isset($ParametersArray['email']) ?
 $ParametersArray['email'] : null).
(isset($ParametersArray['street']) ?
 $ParametersArray['street'] : null).
(isset($ParametersArray['street_n1']) ?
 $ParametersArray['street_n1'] : null).
(isset($ParametersArray['street_n2']) ?
 $ParametersArray['street_n2'] : null).
(isset($ParametersArray['state']) ?
 $ParametersArray['state'] : null).
(isset($ParametersArray['addr3']) ?
 $ParametersArray['addr3'] : null).
(isset($ParametersArray['city']) ?
 $ParametersArray['city'] : null).
(isset($ParametersArray['postcode']) ?
 $ParametersArray['postcode'] : null).
(isset($ParametersArray['phone']) ?
 $ParametersArray['phone'] : null).
(isset($ParametersArray['country']) ?
 $ParametersArray['country'] : null).
(isset($ParametersArray['code']) ?
 $ParametersArray['code'] : null).
(isset($ParametersArray['p_info']) ?
 $ParametersArray['p_info'] : null).
(isset($ParametersArray['p_email']) ?
 $ParametersArray['p_email'] : null).
(isset($ParametersArray['n_email']) ?
 $ParametersArray['n_email'] : null).
(isset($ParametersArray['expiration_date']) ?
 $ParametersArray['expiration_date'] : null).
(isset($ParametersArray['recipient_account_number']) ?
 $ParametersArray['recipient_account_number'] : null).
(isset($ParametersArray['recipient_company']) ?
 $ParametersArray['recipient_company'] : null).
(isset($ParametersArray['recipient_first_name']) ?
 $ParametersArray['recipient_first_name'] : null).
(isset($ParametersArray['recipient_last_name']) ?
 $ParametersArray['recipient_last_name'] : null).
(isset($ParametersArray['recipient_address_street']) ?
 $ParametersArray['recipient_address_street'] : null).
(isset($ParametersArray['recipient_address_building']) ?
 $ParametersArray['recipient_address_building'] : null).
```

```
    (isset($ParametersArray['recipient_address_apartment']) ?
     $ParametersArray['recipient_address_apartment'] : null).
    (isset($ParametersArray['recipient_address_postcode']) ?
     $ParametersArray['recipient_address_postcode'] : null).
    (isset($ParametersArray['recipient_address_city']) ?
     $ParametersArray['recipient_address_city'] : null).
    (isset($ParametersArray['warranty']) ?
     $ParametersArray['warranty'] : null).
    (isset($ParametersArray['bylaw']) ?
     $ParametersArray['bylaw'] : null).
    (isset($ParametersArray['personal_data']) ?
     $ParametersArray['personal_data'] : null).
    (isset($ParametersArray['credit_card_number']) ?
     $ParametersArray['credit_card_number'] : null).
    (isset($ParametersArray['credit_card_expiration_date_year']) ?
     $ParametersArray['credit_card_expiration_date_year'] : null).
    (isset($ParametersArray['credit_card_expiration_date_month']) ?
     $ParametersArray['credit_card_expiration_date_month'] : null).
    (isset($ParametersArray['credit_card_security_code']) ?
     $ParametersArray['credit_card_security_code'] : null).
    (isset($ParametersArray['credit_card_store']) ?
     $ParametersArray['credit_card_store'] : null).
    (isset($ParametersArray['credit_card_store_security_code']) ?
     $ParametersArray['credit_card_store_security_code'] : null).
    (isset($ParametersArray['credit_card_customer_id']) ?
     $ParametersArray['credit_card_customer_id'] : null).
    (isset($ParametersArray['credit_card_id']) ?
     $ParametersArray['credit_card_id'] : null).
    (isset($ParametersArray['blik_code']) ?
     $ParametersArray['blik_code'] : null).
    (isset($ParametersArray['credit_card_registration']) ?
     $ParametersArray['credit_card_registration'] : null).
    (isset($ParametersArray['recurring_frequency']) ?
     $ParametersArray['recurring_frequency'] : null).
    (isset($ParametersArray['recurring_interval']) ?
     $ParametersArray['recurring_interval'] : null).
    (isset($ParametersArray['recurring_start']) ?
     $ParametersArray['recurring_start'] : null).
    (isset($ParametersArray['recurring_count']) ?
     $ParametersArray['recurring_count'] : null);

$ChkValue = hash('sha256',$ChkParametersChain);

if ($Environment == 'production') $EnvironmentAddress =
  'https://ssl.dotpay.pl/t2/';
else if ($Environment == 'test') $EnvironmentAddress =
  'https://ssl.dotpay.pl/test_payment/';

if ($RedirectionMethod == 'POST') {
  $RedirectionCode = '<form action="'.$EnvironmentAddress.'" method="POST"
   id="dotpay_redirection_form">'.PHP_EOL;

  foreach($ParametersArray as $key => $value) {
    $RedirectionCode .= "\t".'<input name="'.$key.'" value="'.$value.
      '" type="hidden"/>'.PHP_EOL;
  }

  $RedirectionCode .= "\t".'<input name="chk" value="'.$ChkValue.
    '" type="hidden"/>'.PHP_EOL;
  $RedirectionCode .= '</form>'.PHP_EOL.
    '<button id="dotpay_redirection_button" type="submit"
     form="dotpay_redirection_form" value="Submit">Confirm and Pay
     </button>'.PHP_EOL;

  return $RedirectionCode;
}
```

```
    else if ($RedirectionMethod == 'GET') {
      $RedirectionCode = $EnvironmentAddress.'?';

      foreach($ParametersArray as $key => $value) {
        $RedirectionCode .= $key.'='.rawurlencode($value).'&';
      }

      $RedirectionCode .= 'chk='.$ChkValue;

      return $RedirectionCode;
    }
  }

// ------------------------------------------------------------------------

  $ParametersArray = array (
    "api_version"  => "dev",
    "amount"  => "15.07",
    "currency"  => "PLN",
    "description"  => "Platnosc za zamowienie 567915976",
    "url"  => "https://www.example.com/thanks_page.php",
    "type"  => "0",
    "buttontext"  => "Wroc do www.example.com",
    "urlc"  => "https://www.example.com/urlc_receiver.php",
    "control"  => "MXdvR1MzaUdLQWRk",
    "firstname"  => "Jan",
    "lastname"  => "Nowak",
    "email"  => "jan.nowak@example.com",
    "street"  => "Warszawska",
    "street_n1"  => "1",
    "city"  => "Krakow",
    "postcode"  => "12-345",
    "phone"  => "123456789",
    "country"  => "POL"
  );

  echo GenerateChkDotpayRedirection ("123456",
    "6aR8J24F3x80Q3MDwrAYGcNm6ReS426y", "test", "POST", $ParametersArray);

?>
```

III. 2. Faulty redirections from the merchant's system

On Dotpay's site a shop may be configurated in such a way that in case of a faulty redirection to Dotpay (caused by e.g. a faulty parameters passed by the merchant's system) an automated redirection will be made to the address from the `url` parameter (sent by merchant's system) along with a faulty code.

When an error occurs a redirection to the address from `url` parameter will be made. `error_code` parameter with a proper value will be added. For example, while redirecting from a merchant's system

```
url=http://example.com/
```

with an unknown payment channel, the result will be redirect to:

```
http://example.com/?error_code=UNKNOWN_CHANNEL
```

`error_code` values are listed below.

`PAYMENT_EXPIRED` - payment link expired or datetime defined in `expiration_date` parameter have been exceeded

`UNKNOWN_CHANNEL` - unknown channel

`DISABLED_CHANNEL` - payment channel stopped

`BLOCKED_ACCOUNT` - blocked account

`INACTIVE_SELLER` - account inactive – payment is not possibile

`AMOUNT_TOO_LOW` - less amount than specified for a shop

`AMOUNT_TOO_HIGH` - higher amount than specified for a shop

`BAD_DATA_FORMAT` - a bad data format has been sent e.g. `expiration_date` wrong format

`UNKNOWN_ERROR` - returned value in other case than the above mentioned

In order to activate functionality which is specifying a proper shop configuration on Dotpay's site a request should be sent to tech@dotpay.pl

IV. TEST ENVIRONMENT

It is possible to create a test environment (independently on production environment) which allows to make a simulation payment (transaction) in order to test the shop integration with doptay payment system.

In order to receive more information concerning test environment please contact Customer Service at bok@dotpay.pl .

The address of a test environemt has been placed below. Any action which should be taken in order to integrate with Dotpay payment system is analogical to the above described.

https://ssl.dotpay.pl/test_payment/ - address of the payment form

https://ssl.dotpay.pl/test_seller/ - address of the administration panel

V. ADDITIONAL INFORMATION

Dotpay IP address:

195.150.9.37

Using the above-mentioned IP service Dotpay sends all confirmation to the merchant's system. For proper communication, the merchant should accept connections from this address.

WARNING !

Merchnat's system should always verify IP address from which a confirmation has been sent for security reasons.

VI. ATTACHMENT 1 - PAYMENT CHANNELS

In the following table all channels which are available in Dotpay system are presented. In the table the following are presented channel names, their providers and numbers, logotypes and remarks

Channel numbers are values which are accepted by `channel` parameter described in the above chapters.

Table 4. (Payment channels available in Dotpay system)

NUMBER	NAME	PROVIDER	LOGO	REMARKS
		PAYMENT CARDS		
71	MasterPass	First Data Polska S.A.	MasterPass	Available only for business accounts.
246	Payment cards via Payeezy	First Data Polska S.A.	VISA MasterCard	Available only for business accounts.
248	Payment cards	-	VISA MasterCard	Available only for business accounts. Needed separate agreement.
		FAST TRANSFERS (wire transfer data are passed directly to the bank)		
1	mTransfer	mBank S.A.	mBank mTRANSFER	
2	Płacę z Inteligo	Inteligo	płacę z inteligo	
4	Płacę z iPKO	Bank PKO BP	płacę z iPKO	
6	Przelew24	Bank Zachodni WBK S.A.	Przelew 24 Bank Zachodni WBK	
18	Przelew z BPH	Bank BPH S.A.	PRZELEW Z Bank BPH	
36	Pekao24Przelew	Bank Pekao S.A.	Bank Pekao Pekao24Przelew	
38	Płać z ING	ING Bank Śląski S.A.	PŁAĆ Z ING	

44	Millennium - Płatności Internetowe	Millennium Bank S.A.		
45	Pay with Alior Bank	Alior Bank S.A.		
46	Płacę z Citi Handlowy	Citi Bank Handlowy S.A.		
48	R-Przelew	Raiffeisen Bank Polska S.A.		
50	Pay Way Toyota Bank	Toyota Bank Polska		
51	Płać z BOŚ	BOŚ Bank S.A.		
56	eurobank - płatność online	Eurobank		
58	Szybkie Płatności Internetowe z Deutsche Bank PBC	Deutsche Bank PBC S.A.		
60	T-Mobile Usługi Bankowe	Alior Bank S.A. Oddział T-Mobile Usługi Bankowe		
63	Płacę z IKO	Bank PKO BP		
65	Paylink Idea Bank	Idea Bank S.A.		
66	Płacę z PBS	Podkarpacki Bank Spółdzielczy		
70	Pocztowy24	Bank Pocztowy S.A.		
72	Płacę z Orange	mBank S.A.		
73	BLIK	Polski Standard Płatności Sp. z o.o.		
74	Banki Spółdzielcze	Krajowa Izba Rozliczeniowa S.A.		

75	Płacę z Plus Bank	Plus Bank S.A.		
76	Getin Bank PBL	Getin Noble Bank S.A.		
80	Noble Pay	Getin Noble Bank S.A.		
81	Idea Cloud	Idea Bank S.A.		

TRANSFERS

(wire transfer data are not passed to the bank. A payer needs to copy or write the data manually)

7	ING Corporate customers	ING Bank Śląski S.A.		
10	Millennium Corporate customers	Millennium Bank S.A.		
16	Credit Agricole	Credit Agricole Bank Polska S.A.		
27	BGZ	Bank BGŻ S.A.		
32	BNP Paribas	BNP Paribas		
33	Volkswagen Bank	Volkswagen Bank Polska		

CASH

11	Bank transfer / postal	-		
21	VIA - Moje Rachunki	BillBird S.A.		
31	Zaplac w Zabce i we Freshmarket	Żabka Polska sp. z o.o.		
35	Kantor Polski	Kantor Polski S.A.		
82	SEPA transfer	-		Available only for business accounts.

				Needed separate agreement.

PURSES AND VOUCHERS

24	mPay	mPay S.A.		
52	SkyCash	SkyCash Poland S.A.		

INSTALLMENTS

55	erata - dotpay installment	Alior Bank S.A.		Available for business accounts only, for payments PLN 300 - PLN 20000.
68	mRaty	mBank S.A.		Available for business accounts only, for payments PLN 300 - PLN 20000.

OTHER

212	PayPal	PayPal		Available only in PLN.

POSTPONED PAYMENTS

77	FerBuy	FerBuy Poland Sp. z.o.o.		Available only for business accounts.

VII. ATTACHMENT II - OPERATION STATUSES DESCRIPTION

In the following table are presented description of operation statuses which are created in system.

STATUS	MEANING / DESCRIPTION
`new`	A new operation. It means establishment of the operation.
`processing`	The operation is being processed, e.g. for *payment* type it means that a payer came back to Dotpay from a payment channel provider or a provider has informed Dotpay about this fact. *Payment* and *payment_multimerchant_child* operations don't have the accounting in the "processing" state. Complaint operation in this state means that a complaint procedure has been started.
`completed`	For payment and *payment_multimerchant_child* types it means that Dotpay has already had (or is sure to have) funds from the payment made by a customer. For *release_rollback* type it means that locked funds from *rollback* were released to the shop's account (ID). For *payout*, *refund* and *complaint* types it means that funds have been transferred to a merchant, to a payer or a payment channel provider. This is the final status which means the operation will not change its status. For this operation additional accounting may be written but only those that are as a result of e.g. correction or correction incorrectly calculated.
`rejected`	For *payment*, *payment_multimerchant_child* types it means that Dotpay has NOT received funds from a payer and is sure that won't receive those funds. This means that e.g. a payer has cancelled a payment or doesn't have enough funds at a payment channel provider (e.g. bank) to execute the payment. For *payout*, *refund* and *complaint* types it means that the operation was cancelled - funds were refunded to a shop's account (ID). This is the final status which means the operation will not change its status. To this operation additional accounting may be written but only those that are as a result of e.g. correction or correction incorrectly calculated.
`processing_realization_waiting`	Operation in this state expects execution, e.g. for *payout* operation it means a withdrawal request (manually by a merchant or by auto withdrawal mechanism). Operation in this state expects execution in "n" business days where "n" is specified in the agreement signed by a merchant.
`processing_realization`	This status means start of withdrawal execution procedure, for *payout* type operation it means payout transfer to the merchant's account, for *refund* type operation funds transfer to a payer's account. If possible (e.g. card payment channels), instead of funds transfer, a refund is executed on a given channel.

CHANGE LOG

VERSION	DATE	CHANGES DESCRIPTION
1.29.11.1	2016-03-21	– adding chapter „ADDITIONAL FEATURES” – adding subchapter „Redirection parameters integrity protection (CHK)” – moving subchapter „Faulty redirections from the merchant’s system” to chapter „ADDITIONAL FEATURES”
1.29.8.1	2016-02-26	– adding parameters „channel_groups” – adding information about „HTTPS verify” and „SSL certificate verify” in “II. RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)” chapter – adding channel 82 („SEPA transfer”) – adding channel 248 („Payment cards”) – channel 64 (“PeoPay”) removal
1.28.5.2	2016-01-07	– adding parameters „ blik_code” – adding chapter “ATTACHMENT II - OPERATION STATUSES DESCRIPTION”
1.28.5.1	2015-12-31	– channel 79 („Open Pay”) removal
1.27.0.1	2015-10-30	– channel 25 („Plus Bank”) removal
1.25.7.1	2015-10-30	– channel 49 („MeritumBank”) removal
1.25.3.1	2015-10-22	– channel 22 („Ukash”) removal – channel 43 („Bank Spółdzielczy we Wschowie”) removal – channel 62 („DNB Nord”) removal
1.25.1.1	2015-10-19	– change of the main payment site (from https://ssl.dotpay.pl to https://ssl.dotpay.pl/t2/) – channel 3 („MultiTransfer”) removal
1.24.9.1	2015-10-12	– adding channel 81 („Idea Cloud”)
1.23.13.3	2015-09-24	– typographical correction
1.23.13.2	2015-08-20	– channel 69 („V.me”) removal
1.23.13.1	2015-08-12	– adding channel 79 („Open Pay”) – adding channel 80 („Noble Pay”) – channel 15 („iPKO”) removal
1.23.9.2	2015-07-30	– removal of UNKNOWN_ACCOUNT value of error_code parameter described in "FAULTY REDIRECTIONS FROM THE MERCHANT’S SYSTEM" chapter
1.23.9.1	2015-07-20	– adding channel 77 („FerBuy”)
1.22.9.1	2015-06-01	– adding channel 74 (“Banki Spółdzielcze”) – adding channel 75 (“Płacę z Plus Bank”)

		– adding channel 76 (“Getin Bank PBL”) – channel 17 (“Płacę z iPKONET”) removal – channel 57 (“Getin Bank”) removal – adding parameters in URLC notifications: „operation_withdrawal_amount”, „operation_commission_amount”, „channel_country”, „geoip_country”
1.20.9.2	2015-02-09	– adding channel 73 (“BLIK”)
1.20.9.1	2015-01-14	– channel 245 (“MasterCard Mobile”) removal – adding chapter “CHANGE LOG”
1.19.15.2	2014-12-12	– adding channel 72 (“Płacę z Orange”)
1.19.15.1	2014-12-08	– adding channel 71 (“MasterPass”)
1.18.5.4	2014-11-19	– adding channel 69 (“V.me”)
1.18.5.3	2014-11-04	– channels 14 (“KB24”) and 61 (“Bank Pocztowy”) removal
1.18.5.2	2014-10-28	– adding channels 66 (“Płacę z PBS”) and 70 (“Pocztowy24”) – renaming channel 17 (from “Płać z Nordea” to “Płacę z IPKOnet”) – adding https://ssl.dotpay.pl/test_seller/ address to “TEST ENVIRONMENT” chapter