



**Dotpay**  
**Technical manual for payments**  
**implementation**

**version: 1.78.22.1**

**en**

**Dotpay Development Team**

Created on : 26 kwietnia 2021

*This manual is also available online in [html format](#)*

# Spis treści

<b>1 Basic version</b>	<b>1</b>
1.1 Example forms of payments / donations	2
<b>2 Advanced version</b>	<b>8</b>
2.1 Diagram 1. Example of the payment process in the following diagram and description:	9
2.2 Table 1. (Basic parameters)	10
2.3 Table 2. (Additional parameters)	11
<b>3 RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)</b>	<b>22</b>
3.1 Table 3. (Parameters sent by Dotpay back to merchant's system)	23
3.2 An example of calculating the signature parameter	31
<b>4 ADDITIONAL FEATURES</b>	<b>32</b>
4.1 Redirection parameters integrity protection (CHK)	32
4.1.1 Basic Example	34
4.1.2 Full Example	37
4.2 One-click and recurring payments	45
4.2.1 Table 4. (Parameters for card registration)	45
4.2.2 Table 5. (Parameters for consecutive payments with previously registered card)	46
4.2.3 Table 6. (Common parameters for both first and consecutive payments)	46
4.2.4 Exemplary integration models and their corresponding requirements)	47
Integration model: 1 click	48
Integration model: recurring	48
Mixed integration model: 1 click + recurring	48
4.2.5 Table 11. (Parameters for credit card data)	51
4.3 Split payment (Multimerchant)	52
4.4 Masscollect	54
4.4.1 Table 12. (Additional parameters used in the Masscollect service)	54
4.5 Shipping and payer data	56
4.5.1 Table 13. (Data handled by customer parameter)	56
4.5.2 Values used for indicator field replacement for selected fields:	58
4.5.3 Dictionary of article categories for the parameter order.items.category	60
4.5.4 Data required in customer parameter for <b>PayPo</b> channel	61
4.5.5 Data required in customer parameter for <b>Raty Alior</b> channel	62
4.5.6 Data required in customer parameter for <b>PayPal</b> channel (only with PayPal SPP)	64
4.6 Faulty redirections from the merchant's system	65
<b>5 TEST ENVIRONMENT</b>	<b>66</b>
5.1 Table 14. (List of sample card numbers that can be used in the process of testing card payments)	67
<b>6 ADDITIONAL INFORMATION</b>	<b>67</b>
6.1 Verification of the IP address	67
6.2 Security of payment integration	67
6.2.1 Redirecting customer from shop to Dotpay	68
6.2.2 Receiving confirmation of the posted payment in Dotpay	68
<b>7 ATTACHMENT I - PAYMENT CHANNELS</b>	<b>69</b>
7.1 Table 15. (Payment channels available in Dotpay system)	69
7.1.1 PAYMENT CARDS	69
7.1.2 FAST TRANSFERS	69
7.1.3 TRANSFERS	72

7.1.4 CASH . . . . .	72
7.1.5 PURSES AND VOUCHERS . . . . .	73
7.1.6 INSTALLMENTS . . . . .	73
7.1.7 OTHERS . . . . .	74
7.1.8 POSTPONED PAYMENTS . . . . .	74
7.1.9 MOBILE TRANSFERS - Direct Carrier Billing type . . . . .	74
7.2 Payment Channel List - available for download . . . . .	75
<b>8 ATTACHMENT II - OPERATION STATUSES DESCRIPTION</b>	<b>76</b>
8.1 Table 16. (Operation statuses created in the Dotpay system) . . . . .	76
<b>9 CHANGELOG</b>	<b>77</b>
<b>HTTP Routing Table</b>	<b>82</b>

## Spis treści

---

There are two methods of accepting payments from customers. The first method (presented in subchapter Basic version) is intended for Sellers who do not have e-shops or services integrated with the Dotpay system.

The second method (presented in subchapter Advanced version) is designed for e-shops and services, at which the payment process begins (e.g. customer is ordering some item).

Dotpay payment gateway is located at:

- production environment: <https://ssl.dotpay.pl/t2/>
- test environment : [https://ssl.dotpay.pl/test\\_payment/](https://ssl.dotpay.pl/test_payment/)

**Caution:** Because of security policy, Dotpay payment gateway cannot be embedded in iframe (X-Frame-Options:SAMEORIGIN).

## 1 Basic version

To redirect the customer to payment, all data can be saved in the URL - link. In the example below, link\_pre-id01| id sklepu, amount , currency and description is passed to the payment form:

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

The above address redirects by the method: *GET*.

Other parameters that can be passed in this way to Dotpay payment form are listed in [Table 1](#) and [Table 2](#) of this document.

After logging 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 lang .. Generating a payment request means creating a special key which allows defining the given parameters thus prevent modification of payment data.

Links 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 payment language was **NOT** specified or defined language was removed, parameter lang 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>

Additionally, you can attach the ignore\_last\_payment\_channel = 1 parameter to the payment link in order to ignore the payment method recently chosen by the customer (stored in the browser's memory), so that he will always see the full list of available payment methods.

[https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbyj6t17&ignore\\_last\\_payment\\_channel=1&lan](https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbyj6t17&ignore_last_payment_channel=1&lan)

Such a prepared link with a key can be sent for example by e-mail to a buyer.

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

**Caution:** By default for shop is also required sending the `chk` parameter. In the case of generating payment links from seller administration panel the `chk` parameter is automatically added to the link.

However for links generated manually or using seller administration panel API `chk` value has to be calculated and added separately.

Exemplary link is going to look like:

`https://ssl.dotpay.pl/t2/?chk=c7dc9bb5d0c726a44cf478d3b78390011641f0f52c3db3c0ba6b9a658c6fb`

How to calculate `chk` has been described in *Redirection parameters integrity protection (CHK) section*. .

---

**Note:** If shop doesn't use automatic notifications *URLC* and confirmation of each transaction is carried out after manual verification of correctness of booked amount and currency for a specific order, the maturity of the `chk` parameter may be disabled. For this purpose please contact us by email at the address [administracja@dotpay.pl](mailto:administracja@dotpay.pl)

---

## 1.1 Example forms of payments / donations

**Caution:** For proper functioning below forms require prior modification of parameters specified in section *CONFIGURATION*, for example inserting appropriate account `id` in which payment is to be made (`id` number can be found after logging in to the seller administration panel tab *Settings*, this is a 6-digit number placed after the sign `#` in column *Shop*). Otherwise, you will be redirected to the page with error message.

Customer after clicking button is redirected to the Dotpay payment form (`https://ssl.dotpay.pl/t2/`) to choose a payment method. Simultaneously with the *POST* method are sent the following parameters:

- a) seller's ID (e.g. `id = 123456`)
- b) transaction's amount (e.g. `amount = 12.42`)
- c) currency
- d) transaction's description (e.g. `description = Payment for invoice #12345/2014`)
- e) redirection's type (e.g. `type = 0`)
- f) return address after payment (e.g. `url = https://www.example.com`)
- g) return button text (e.g. `buttontext = Return`)

**Caution:** The following forms don't include a function to calculate the mandatory `chk` parameter. This functionality should be completed on its own or contact Dotpay administration department ([administracja@dotpay.pl](mailto:administracja@dotpay.pl)) in order to disable the verification of `chk`. Please read the *Security of payment integration* chapter before.

The form may also send other, optional parameters of which description can be found in the tables in the further parts of documentation.

### 1. Form with predefined and any amount<sup>1</sup>

```
1 <html>
2
3 <head>
4   <meta charset="UTF-8">
5 </head>
6 <!-------
7
8   Górna część strony
9
10 ----->
11
12 <!--- copy start-->
13
14 <script type="text/javascript" src="http://code.jquery.com/jquery-3.5.1.
15 ↪min.js"></script>
16
17 <style>
18   div.dotpay_form_donation {
19     font-family: sans-serif;
20     text-align: center;
21   }
22
23   div.dp_temat {
24     font-size: 1.5em;
25     font-style: inherit;
26     font-weight: bold;
27     color: #334242;
28   }
29
30   input#dp_def_amount {
31     border: 1px solid #bbb;
32     border-radius: 3px;
33     height: 50px;
34     font-size: 1.3em;
35     background: #dae6ff;
36     text-align: center;
37     font-weight: 500;
38     cursor: pointer;
39   }
40
41   input#dp_kwota {
42     border: 1px solid #bbb;
43     border-radius: 3px;
44     font-size: 1.2em;
45     background: #f8f6fb;
46     text-align: center;
47
48   }
49
50   input#dp_other_amount {
```

(continues on next page)

<sup>1</sup> [https://www.dotpay.pl/developer/doc/api\\_payment/pl/files/code/form/form1.html](https://www.dotpay.pl/developer/doc/api_payment/pl/files/code/form/form1.html)

```

51     font-size: 1em;
52     background: #daedff;
53     border: 1px solid #bbb;
54     border-radius: 3px;
55     padding: 5px;
56     text-align: center;
57     cursor: pointer;
58 }
59
60     button#dp_buttonDarowizna {
61     font-size: 1.3em;
62     background: #ae3131;
63     border: 1px solid #bbb;
64     border-radius: 3px;
65     padding: 5px;
66     text-align: center;
67     cursor: pointer;
68     color: #f3f0ed;
69     letter-spacing: 0.1em;
70 }
71     table.tbl_center{
72     margin-left: auto;
73     margin-right: auto;
74 }
75
76 </style>
77
78 <script>
79     $(document).ready(function () {
80         $("#dp_buttonDarowizna").click(function () {
81             if ($('#dp_kwota').val().trim() === '') {
82                 $("#dp_kwota_alert").text("Proszę wybrać lub wprowadzić
83 →kwotę darowizny.").show();
84                 $("#dp_kwota_alert").css("display", "inline").
85 →fadeOut(5000);
86                 return false;
87             }
88         });
89     });
90 </script>
91
92 <div class="dotpay_form_donation">
93     <form action="https://ssl.dotpay.pl/t2/" method="post" target="_
94 →parent">
95         <div class="dp_temat">Wybierz kwotę darowizny</div>
96         <p>
97             <input type="button" id="dp_def_amount"
98             onClick="$($('#dp_kwota').val('10')); $('#dp_kwota').prop(
99 →'readonly', true);$('#dp_kwota').attr('style','color:blue');$('#dp_other_
100 →amount_txt').html('Wybrana Kwota')"
101             value="10.00 zł" />
102             <input type="button" id="dp_def_amount"
103             onClick="$($('#dp_kwota').val('20')); $('#dp_kwota').prop(
104 →'readonly', true);$('#dp_kwota').attr('style','color:blue');$('#dp_other_
105 →amount_txt').html('Wybrana Kwota')"
106             value="20.00 zł" />

```

(continues on next page)

```

100     <input type="button" id="dp_def_amount"
101         onClick="$('#dp_kwota').val('50'); $('#dp_kwota').prop(
↳ 'readonly', true); $('#dp_kwota').attr('style','color:blue'); $('#dp_other_
↳ amount_txt').html('Wybrana Kwota')"
102         value="50.00 zł" />
103     <input type="hidden" name="type" value="0" />
104     <input type="hidden" name="currency" value="PLN" />
105
106     <table class="tbl_center">
107         <tr>
108             <td>
109                 <br><input type="button" id="dp_other_amount"
110                 onClick="$('#dp_kwota').prop('readonly',
↳ false); $('#dp_kwota').attr('style','color:brown'); $('#dp_other_amount_txt
↳ ').html('<span style=\`color:brown\`>Wprowadź kwotę</span>')"
111                 value="Inna kwota" />
112             </td>
113             <td>
114                 <br><span id="dp_other_amount_txt">Wybrana Kwota
↳ </span>:
115                 <input type="text" name="amount" id="dp_kwota"
↳ size="5" readonly
116                 pattern="^([1-9])((\.\d{1,2})?)$|^((?!0)(\d
↳ {1,5})((\.\d{1,2})?)$|^(1(\d{5})(\.\d{1,2})?)$|^(200000(. [0]{1,2})?)$"
117                 placeholder="np. 100" maxLength="9" size="9"
118                 title="Kwota powinna zawierać się w
↳ przedziale 1 - 200000 PLN. Dozwolony format to np: 100 lub 152.43"
119                 oninput="this.value = this.value.replace(/[\^
↳ 0-9\.\]/g, ''); this.value = this.value.replace(/(\.\.)*\./g, '$1');" />
120                 PLN<br />
121             </td>
122         </tr>
123     </table>
124
125     <!------- KONFIGURACJA ----->
126     <!------->
127
128     <!--- zamiast 000000 należy podstawic numer ID w Dotpay -->
129     <input type="hidden" name="id" value="000000" />
130     <!--- Tytuł transakcji ---->
131     <input type="hidden" name="description" value="Testowa
↳ płatność" />
132     <!--- Ardes URL powrotu ---->
133     <input type="hidden" name="url" value="http://www.example.com
↳ " />
134     <!-- Tekst przycisku powrotu do sklepu ---->
135     <input type="hidden" name="buttontext" id="buttontext" value=
↳ "Powrót do sprzedawcy" />
136     </p>
137
138     <!------- KONIEC KONFIGURACJI ----->
139     <!------->
140     <p><br><button class="dp_buttomDarowizna" id="dp_buttomDarowizna
↳ ">Wpłać darowiznę</button></p>

```





```

35 <!-------
36
37     Dolna część strony
38
39 ----->
40
41 </html>

```

### 3. Form with any description and amount<sup>3</sup>

```

1 <html>
2 <meta charset="utf-8">
3 <!-------
4
5     Górna część strony
6
7 ----->
8
9 <div style="text-align: center;">
10 <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
11 <p style="font-size: 18px">Wybierz kwotę darowizny</p>
12 <input name="description" value="Darowizna na cele statutowe" type=
   ↳ "hidden">
13 <input name="amount" id="kwota" size="6" value="" type="text" required
14     pattern="^([1-9])((\.\d{1,2})?)$|^((?!0)(\d){1,5})(\.\d{1,2})?)$|^
   ↳ (1(\d{5})(\.\d{1,2})?)$|^(200000(\.[0]{1,2})?)$"
15     placeholder="np. 10" maxlength="9" size="9"
16     title="Kwota powinna mieścić się w przedziale 1 - 200 000 PLN.↳
   ↳ Dozwolony format to np: 10 lub 10.00">PLN
17
18 <!------- KONFIGURACJA -----
   ↳ ----->
19
20 <!------ zamiast 000000 należy podstawic numer ID w Dotpay -->
21 <input name="id" value="000000" type="hidden">
22
23 <!------- KONIEC KONFIGURACJI -----
   ↳ ----->
24
25 <p><br><button class="buttonDarowizna">Wpłać darowiznę</button></p>
26 </form>
27 <br>
28 </div>
29
30 <!-------
31
32     Dolna część strony
33
34 ----->
35
36 </html>

```

<sup>3</sup> [https://www.dotpay.pl/developer/doc/api\\_payment/pl/files/code/form/form3.html](https://www.dotpay.pl/developer/doc/api_payment/pl/files/code/form/form3.html)

## 2 Advanced 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/*.

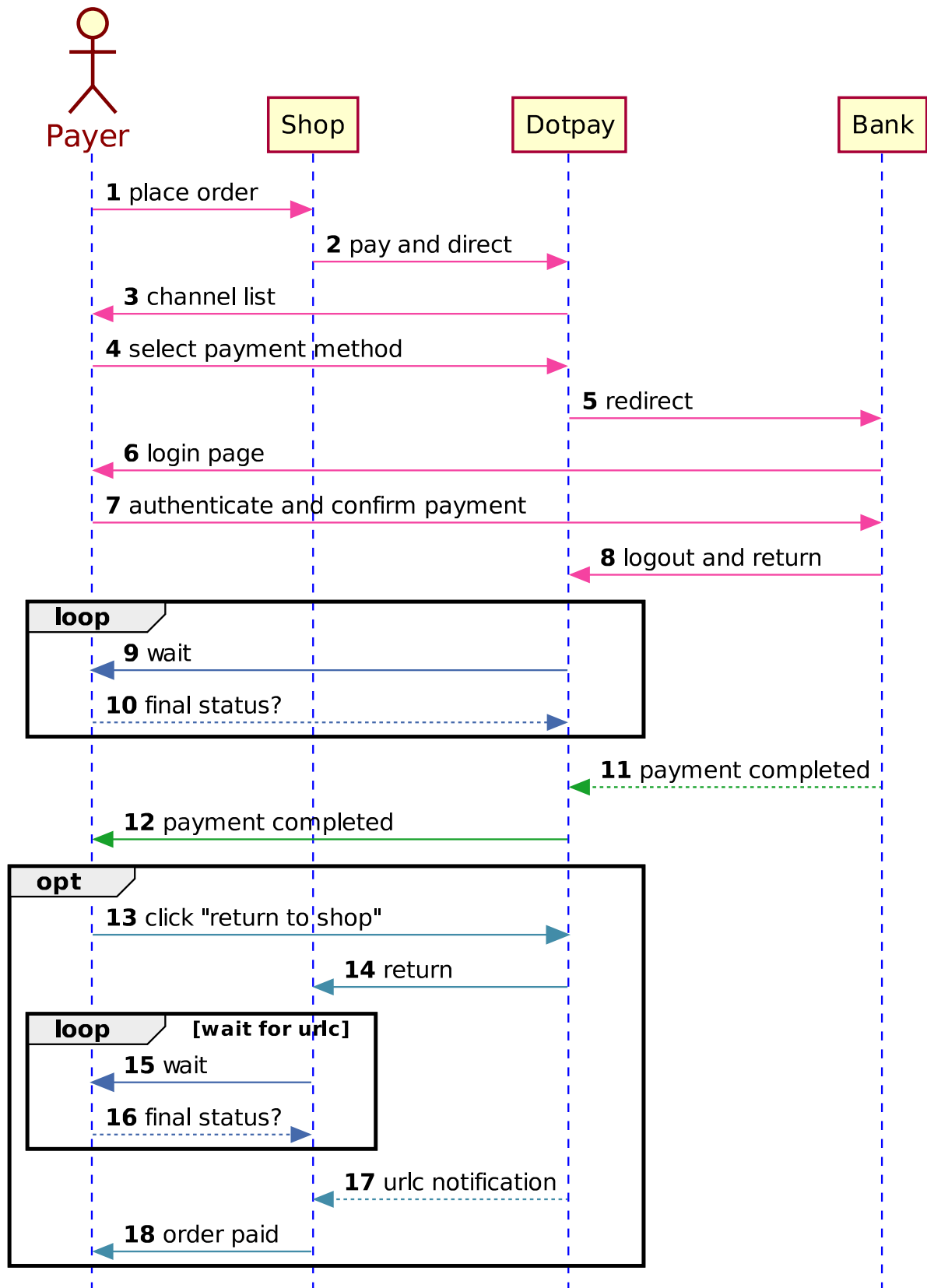
In the next step the customer makes the payment with use of selected payment channel. If there is a defined return address and right parameter value of type , on the payment confirmation page will be displayed a button to return to the seller web page.

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 (*Settings -> Notifications -> URLC Configuration -> Edit*) or sent when you initiate the payment as a parameter named *urlc* .

In the second case you should unlock for your store getting the *urlc* parameters from external sources, ie deselect the option: *Block external urlc (Settings -> Notifications -> URLC Configuration -> Edit)* . With such a configuration a seller can define other *URLC* addresses for each payment.

The principle of *URLC* operation notifications is described in chapter *URLC CONFIRMATION*

**2.1 Diagram 1. Example of the payment process in the following diagram and description:**



1. Buyer places an order in shop

2. after completing shopping cart customer selects payment method with Dotpay and is redirected to the payment format
3. where presented is channel list.
4. Customer makes a choice
5. and is redirected to the bank.
6. The login page for banking is shown
7. where he enters credentials and confirms the transfer.
8. After logging out is following return to the Dotpay website.
9. During waiting for confirmation
10. the browser periodically inquires about the status of payment.
11. Bank informs Dotpay about the final status of payment
12. and this information is presented to payer.

---

**Note:** Steps 13 - 18 are optional (depending on the model of integration) and don't have effect on payment process.

---

13. Buyer clicks on return button
14. and is redirected to the store
15. During waiting for confirmation
16. the browser periodically inquires about the status of payment.
17. After receiving URLC notification
18. shop confirms that customer has paid for order.

## 2.2 Table 1. (Basic parameters)

PARAMETER	MEANING / DESCRIPTION
api_version	<p>API version - according to which the system sends <i>URLC</i> confirmation (see chapter <i>RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)</i>).</p> <p>Sending parameter overrides default shop ( id ) configuration from seller administration panel in <i>Settings -&gt; Shop configuration -&gt; Edit tab</i>.</p> <p>Available values:  dev - API version described in this manual.</p> <hr/> <p><b>Important:</b> If shop configuration (<i>Settings -&gt; Shop configuration -&gt; Edit tab</i>) has API version set to something else than dev, not sending this parameter will cause Dotpay system to send no longer developed version of <i>URLC</i> notification, which is not compliant with this documentation.</p> <hr/> <p>Example:  api_version = dev</p>

Continued on next page

Tablica 1 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
id	Account ID for which payment is made. ID can be found in the administration panel in <i>Settings</i> tab, it is a 6-digit number after # <i>Shop</i> . column. type: <i>integer</i> min value: 1 max value: 999999 Example: id = 123456
amount	The amount of the transaction given with centesimal part (always two places after the separator). Centesimal separator is a dot (.) type: <i>string</i> min length: 1 max length: 10 exemplary regular expression (for amount in the range 0.01 - 200000.00) : $\wedge\backslash.(0)([1-9])\$ \wedge\backslash.((([1-9])\backslash(d)?)\$ \wedge([1-9])(\backslash.\backslashd\{1,2\})?)\rightarrow)\$ \wedge((?!0)\backslashd\{1,5\})(\backslash.\backslashd\{1,2\})?\$ \wedge(1\backslashd\{5\})(\backslash.\backslashd\{1,2\})?\rightarrow)\$ \wedge(200000(\backslash.\backslashd\{1,2\})?)\$$ Example: amount = 42.82
currency	This parameter determines the currency in which the amount parameter is given. Format according to ISO 4217 <sup>4</sup> standard. Available values: PLN, EUR, USD, GBP, JPY, CZK, SEK, UAH, RON, NOK, BGN, CHF, HRK, HUF, RUB Example: currency = EUR
description	Payment description. type: <i>string</i> min length: 1 max length: 255 Example: description = Invoice 120/2018
chk	Checksum used to verify sent data integrity. Description of functionality can be found in <i>Redirection parameters integrity protection (CHK)</i> section. <hr/> <b>Important:</b> Required by default. <hr/>

### 2.3 Table 2. (Additional parameters)

<sup>4</sup> [https://en.wikipedia.org/wiki/ISO\\_4217](https://en.wikipedia.org/wiki/ISO_4217)

PARAMETER	MEANING / DESCRIPTION
channel	<p>This parameter defines which payment channel will be selected after redirecting client to Dotpay payment form. Numerical values only available (see <i>ATTACHMENT 1 - PAYMENT CHANNELS</i>)</p> <p>Example: channel = 1</p> <p>Seller wishes 1 to select a mTransfer channel by default. List of available payment channels for given shop (id) might be downloaded using:</p> <ul style="list-style-type: none"> <li>• <a href="#">Seller administration panel API<sup>5</sup></a></li> <li>• Payment API (this mechanism is also used by widget, which is a tool to display available payment channels. Is has been described in administration panel in <i>Tools -&gt; Widget</i> tab )</li> </ul> <p>Resource addresses: <a href="https://ssl.dotpay.pl/t2/payment_api/channels/">https://ssl.dotpay.pl/t2/payment_api/channels/</a>  Required parameters: id , amount , currency  Optional parameters: lang , format (json lub xml)</p> <p>Example:  <a href="https://ssl.dotpay.pl/t2/payment_api/channels/?id=123456&amp;amount=">https://ssl.dotpay.pl/t2/payment_api/channels/?id=123456&amp;amount=</a></p>
ch_lock	<p>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.</p> <p>Available values:  0 - channel will not be forced (default)  1 - force the selected channel .</p> <p>Example:  Sending parameters channel = 1 and ch_lock = 1 will not allow a customer to choose other payment method than mTransfer.</p>
ignore_last_payment_channel	<p>Sending parameter will cause Dotpay website to ignore last payment channel used by costomer (saved in their browser data), i.e. all available payment channels will be shown as if it was first payment.</p> <p>By default for consecutive payments last used channel is shown with an option to choose another one.</p> <p>Available values:  1- ignore last chosen channel</p> <p>Example:  ignore_last_payment_channel = 1</p>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
channel_groups	<p>This parameter defines which channel groups will be displayed after redirecting client to Dotpay payment form (by default all are presented). When using this parameter it is recommended to use ignore_last_payment_channel described above. Available values: K - payment cards, T - fast transfers, P - transfers, G - cash, W - purses and vouchers, R - installments, M - mobile transfers (DCB), O - postponed payments, U - Payment Initiation Service (PIS), I - other.</p> <p>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,I channel_groups = R,I,P</p>
url	<p>This parameter defines the Internet address (<i>HTTP</i> or <i>HTTPS</i>) to which buyer is expected to return after payment. Behavior of url parameter is determined by type parameter. type: <i>string</i> max length: 1000</p> <p>Example: url = https://www.example.com/thanks_page.php url = http://195.150.9.55/thanks_page.php When both url and type = 0 parameters are sent 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 <i>POST</i> and <i>GET</i>), which contains information about the possible errors at the checkout pages ( OK or FAIL).</p> <hr/> <p><b>Important:</b> 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 .</p> <hr/> <p>Example: url = https://www.example.com/thanks_page.php?status=OK</p>

Continued on next page



Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
type	<p>This parameter defines a method of redirection to the seller's web page. Parameter value influences url parameter.</p> <p>Available values:</p> <p>0 - after the payment, the buyer will see a button to return to the seller's web page,                  2 - no reaction, lack of a button (default value).                  4 - a direct redirection to the payment channel provider (e.g. bank) as well as after payment making and logging out of the service (called <b>on-site / white label</b> ). In order for functionality to fully work it is required to send necessary parameters needed for given payment channel. If any of them is NOT sent, seller is going to be redirected to Dotpay page to fill missing information, return process will be unchanged (redirection from channel provider to url will be automatic).</p> <hr/> <p><b>Important:</b> If direct redirection mechanism (type = 4) is used it is necessary to add Dotpay Regulations of Payments (parameter bylaw ) and personal data use for payment processing (parameter personal_data ) agreement checkboxes.</p> <hr/> <p>Example:</p> <p>After sending the given parameters set a button will be displayed allowing a Buyer to redirect to <a href="https://www.example.com/thanks_page.php">https://www.example.com/thanks_page.php</a>:</p> <pre>type = 0 url = https://www.example.com/thanks_page.php</pre> <p>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 from the payment provider service a buyer will be redirected to url address. Using the given value allows a scheme e.g. <i>Shop -&gt; Bank -&gt; Shop</i>:</p> <pre>type = 4 url = https://www.example.com/thanks_page.php bylaw = 1 personal_data = 1</pre>
buttontext	<p>The content that appears on a button to return to the seller website. Default text is Back to shop.</p> <p>type: <i>string</i>                  min length: 4                  max length: 100</p> <p>Example:</p> <pre>buttontext = Back to www.example.com</pre>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
bylaw	<p>Parameter informing client has accepted Dotpay Sp. z o.o. Regulations of Payments and cookies policy. If this parameter is used, seller should display a checkbox to the client with information as follows.</p> <p style="padding-left: 40px;">Akceptuję <a href="https://ssl.dotpay.pl/t2/cloudfs1/magellan_media/regulamin_platnosci">regulamin płatności</a> Dotpay sp. z o.o..</p> <p>Available values: 1 - payment regulations accepted</p> <p>Example: bylaw = 1</p>
personal_data	<p>Parameter informing about displaying to the Payer information on the processing of personal data by Dotpay Sp. z o.o..</p> <p>If this parameter is used, seller should display content similar to the one below:</p> <p style="padding-left: 40px;">Przyjmuję do wiadomości, że w celu realizacji procesu płatności Administratorem moich danych osobowych jest Dotpay sp. z o.o.. (KRS 0000700791), 30-552 Kraków (Polska), Wielicka 28B, +48126882600, <a href="mailto:bok@dotpay.pl">bok@dotpay.pl</a>, zobacz regulamin dotpay.pl/t2/cloudfs1/magellan_rod media/rodo"&gt; pełną treść klauzuli informacyjnej</p> <p>Available values: 1 - information on hearing the clause information</p> <p>Example: personal_data = 1</p>

Continued on next page

Tablica 2 – continued from previous page

PARAMETER	MEANING / DESCRIPTION
urlc	<p>Web address (<i>HTTP</i> or <i>HTTPS</i>) to receive parameters confirming or denying completion of the transaction. A detailed description of <i>URLC</i> confirmation is described further in chapter <i>RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)</i>.</p> <p>It is recommended for the address to lead directly to the receiving file. If any <i>301</i> or <i>302</i> redirections occur in the way (which allow <i>HTTP</i> method to change), notification from Dotpay might be empty.</p> <p>Another way is to use <i>307 / 308</i> redirections on the server, since they do not allow such behavior.</p> <p>Sending parameter overrides default shop ( <i>id</i> ) configuration from seller administration panel in <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit tab</i> .</p> <hr/> <p><b>Important:</b> If option <i>Block external urlc</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) is checked, <i>urlc</i> parameter will be ignored.</p> <hr/> <p><b>Important:</b> If option <i>HTTPS verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) is checked, address has to use <i>HTTPS</i> protocol.</p> <hr/> <p>type: <i>string</i>  max length: <i>1000</i>  Example:  urlc = <a href="https://www.example.com/urlc_receiver.php">https://www.example.com/urlc_receiver.php</a>  urlc = <a href="http://195.150.9.55/urlc_receiver.php">http://195.150.9.55/urlc_receiver.php</a></p>
expiration_date	<p>Payment request expiration date, according to the Europa / Warsaw time zone. When using this parameter it is recommended to sign the request sent to Dotpay. Description of signature creation can be found in <i>Redirection parameters integrity protection (CHK)</i> section.</p> <p>Format: <i>YYYY-MM-DD hh:mm:ss</i></p> <p>Example:  expiration_date = 2019-06-01 12:06:37</p>
control	<p>Control parameter allows you to store sequence of data. Unchanged parameter is sent to the Merchant's service in <i>URLC</i> confirmation.</p> <p>type: <i>string</i>  max length: <i>1000</i></p> <p>Example:  control = ec4bf09d3dbe0cb71e6abc3ea44a7273</p>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
firstname	<p>This parameter identifies the first name of the person making the payment.</p> <p>type: <i>string</i>  min length: 1  max length: 50  regular expression:</p> <pre>^\p{L}0-9\s\-\_]{1,50}\$</pre> <p>Example:  firstname = John</p>
lastname	<p>This parameter identifies the surname of the person making the payment.</p> <p>type: <i>string</i>  min length: 1  max length: 50  regular expression:</p> <pre>^\p{L}0-9\s\-\_]{1,50}\$</pre> <p>Example:  lastname = Smith</p>
email	<p>E-mail address of a person making the payment. Payment confirmation will be sent at this address.</p> <p>type: <i>string</i>  max length: 100</p> <p>Example:  email = johnsmith@example.com</p>
street	<p>Street name. A part of the address of the buyer.</p> <p>type: <i>string</i>  min length: 1  max length: 100  regular expression:</p> <pre>^\p{L}0-9\.\s\-\'_,\]{1,100}\$</pre> <p>Example:  street = Wielicka</p>
street_n1	<p>Building number. A part of the address of the buyer.</p> <p>type: <i>string</i>  min length: 1  max length: 30  regular expression:</p> <pre>^\p{L}0-9\s\-\_\/]{1,30}\$</pre> <p>Example:  street_n1 = 4</p>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
street_n2	Flat number. A part of the address of the buyer. type: <i>string</i> max length: 30 regular expression: $\text{^\[}\backslash\text{p}\{L\}0-9\backslash\text{s}\backslash\text{-}\_]\{0,30\}\text{\$}$ Example: street_n2 = 18
state	Additional address data. A part of the address of the buyer. type: <i>string</i> max length: 50 Example: state = NY
addr3	Additional address data. A part of the address of the buyer. type: <i>string</i> max length: 50
city	City name. A part of the address of the buyer. type: <i>string</i> min length: 1 max length: 50 regular expression: $\text{^\[}\backslash\text{p}\{L\}0-9\backslash\text{.}\backslash\text{s}\backslash\text{-}\backslash'\_,\,]\{1,50\}\text{\$}$ Example: city = Kraków
postcode	Postal code. A part of the address of the buyer. type: <i>string</i> max length: 20 regular expression: $\text{^\[}\backslash\text{d}\backslash\text{w}\backslash\text{s}\backslash\text{-}\_]\{0,20\}\text{\$}$ Example: postcode = 30-552
phone	Buyer's phone number. type: <i>string</i> min length: 3 max length: 20 regular expression: $\text{^\[}\backslash\text{+}\backslash\text{s}0-9\backslash\text{-}\_]\{3,20\}\text{\$}$ Example: phone = +48 127654321

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
country	<p>Country name. A part of the address of the buyer. The name should be given in accordance with the <a href="#">ISO 3166-1<sup>6</sup></a> (alfa-2 or alfa-3) or a text English short name (using title case).</p> <p>type: <i>string</i> max length: 50</p> <p>Example: country = PL country = POL country = Poland</p>
lang	<p>Defines the language of web page and payment forms. If parameter is not sent, pages will be displayed based on browser language. If given language is not available, English will be used.</p> <p>Available values: cs - Czech, de - German, en - English, es - Spanish, fr - French, hu - Hungarian, it - Italian, pl - Polish, ro - Romanian, ru - Russian, uk - Ukrainian lt - Lithuanian lv - Latvian</p> <p>Example: lang = en</p>
customer	<p>Recipient and shipping data. Parameter required for some channels (eg. 95 /PayPo/ ). This data should be in <a href="#">JSON<sup>7</sup></a> a format and then encoded using <a href="#">Base64<sup>8</sup></a>. List of data for this parameter and it's encoding has been described in <i>Shipping and payer data</i> section.</p> <p>Value is not returned to merchant anywhere (e.g. in <i>URLC</i> notifications or administration panel).</p> <hr/> <p><b>Important:</b> Using this parameter requires given account ( id ) to have enabled <i>CHK</i> verification.</p> <hr/> <p>type: <i>string</i> Example: customer = eyJyZWdpc3RlcmVhX3NpbmNlIjoiMjAxNy0xMi0zM</p>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
deladdr	<p>Delivery address. Parameter is used by Dotpay system only for informational purposes.</p> <p>Value is not returned to merchant anywhere (e.g. in <i>URLC</i> notifications or administration panel).</p> <p>type: <i>string</i> max length: 500</p> <p>Example: deladdr = Delivery Point 3, Warszawa, ul. Ogonowa 14</p>
p_info	<p>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.</p> <p>Sending parameter overrides default shop ( id ) configuration from seller administration panel in <i>Settings -&gt; Shop configuration -&gt; Edit tab</i></p> <p>type: <i>string</i> max length: 300</p> <p>Example: p_info = Sklep example.com</p>
p_email	<p>This parameter describes an e-mail address, which will be seen by a customer. It overwrites an email address provided during registration process.</p> <p>Sending parameter overrides default shop ( id ) configuration from seller administration panel in <i>Settings -&gt; Shop configuration -&gt; Edit tab</i>.</p> <p>type: <i>string</i> max length: 100</p> <p>Example: p_email = office@example.com</p>

Continued on next page

Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
pid	<p>Payment link for a given store ( id ) generated via Dotpay Admin Panel (in the tab <i>Tools</i> -&gt; <i>Payment links generator</i> -&gt; <i>Generate a payment link</i>) or via the <a href="#">Seller administration panel API</a><sup>9</sup> .</p> <p>type: <i>string</i> length: 32</p> <p>Example: pid = rfhu4jb5ym657g3xluf4bbqfmbj6t17</p> <p>The pid parameter can appear either alone in the payment link, e.g. https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17</p> <p>or with additional parameters: lang , ignore_last_payment_channel , eg: https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&amp;lang=pl&amp;ignore_last_payment_channel=1</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>Caution:</b> By default, the store also requires the parameter link_pre_chk01  chk to be sent. In the case of generating payment links from the administration panel, the chk parameter is automatically added to the link. However, when the payment link is generated manually from parameters or using the administration panel API, it is necessary to generate the correct value chk and attaching it to the payment link.</p> </div>

Continued on next page



Tablica 2 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
blik_code	<p>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.</p> <p>type: <i>string</i>  min length: 6  max length: 6  regular expression:</p> <p style="text-align: center;"><code>^\d{6}\$</code></p> <p>Example:  blik_code = 264230</p>
gp_token	<p>Encoded using the <b>Base64</b> function token received from Google Pay after the merchant's implementation of the <a href="#">Google Pay API<sup>10</sup></a>. It contains encrypted data of the tokenized pay card used to process the payment. Sending the parameter in redirection allows shortening payment process because the authentication of the payer's card data is already on the seller's store page.</p> <p>type: <i>string</i></p> <p>Example:  gp_token = eyJzaWduYXR1cmUiOiJNRVFDZSUZDSm5MQWI1Rk50N...</p>
ap_token	<p>Encoded using the <b>Base64</b> function token received from Apple Pay after the merchant's implementation of the Apple Pay. It contains encrypted data of the tokenized pay card used to process the payment. Sending the parameter in redirection allows shortening payment process because the authentication of the payer's card data is already on the seller's store page/payer device.</p> <p>type: <i>string</i></p> <p>Example:  ap_token = eyJzacmUiOiJ ... h0UXAwM2RcIn0ifQ==</p>

### 3 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

<sup>5</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/en/#shop-payment-channels-list-id](https://www.dotpay.pl/developer/doc/api_seller/en/#shop-payment-channels-list-id)

<sup>6</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1](https://en.wikipedia.org/wiki/ISO_3166-1)

<sup>7</sup> <https://en.wikipedia.org/wiki/JSON>

<sup>8</sup> <https://en.wikipedia.org/wiki/Base64>

<sup>9</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/en/#payment-link-generation](https://www.dotpay.pl/developer/doc/api_seller/en/#payment-link-generation)

<sup>10</sup> <https://www.dotpay.pl/developer/doc/google-pay/en/>

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.

Verifying whether transaction is unique in merchant's system should **NOT** be based only on value of `control == control` parameter. If for some reason customer redirects multiple times between shop and Dotpay or Dotpay and bank, there is a possibility of receiving for example 2 notifications with contradicting status for the same `control`. Transaction should be identified by taking as well `operation_number` into account, transaction id given by Dotpay.

**Caution:** No verification of a `amount == operation_original_amount` , `currency == operation_original_currency` and `signature` parameters on a merchant's site is unsecure and may cause financial loss.

Content and result of *URLC* notifications is available in Dotpay panel by showing given operation details (by clicking on its number) in *Payments -> Operations* list menu. In the same place there is also an option to resend those notifications.

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

*OK*

**(only two capital letters, nothing more - encoding UTF8 without BOM, HTTP response status code 200)**. *OK* is a correctly received and processed confirmation, for example:

```
echo "OK";
```

When a different answer than *OK* is reversed by the Seller's system Dotpay system will repeat confirmations (every few, several minutes). The `X-Dotpay-URLC-Number` parameter may be passed in the notification header, which will be one value higher than the previous one, counting from '0' as the first message.

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

---

**Note:** Optional parameters are not returned by default. To activate them contact Dotpay tech support ( [tech@dotpay.pl](mailto:tech@dotpay.pl) ).

---

---

**Note:** By default system sends notifications *URLC* only for payment operations for completed and rejected status (listed in table below). To enable notifications for other operations and statuses contact Technical Support at [tech@dotpay.pl](mailto:tech@dotpay.pl) .

---

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

PARAMETER	MEANING / DESCRIPTION
id	Account ID. typ: <i>integer</i> min value: 1 max value: 999999 Example: id = 123456
operation_number	Transaction number. Format according to regular expression: $\text{^M}\{4,5\}\{-}\{4,5\}\text{\$}$ Example: operation_number = M1234-56789
operation_type	Transaction type. Available values: payment , payment_multimerchant_child , payment_multimerchant_parent , refund , payout , payout_any_amount , release_rollback , unidentified_payment complaint credit_card_registration payout_commission - commission on payment Example: operation_type = payment
operation_status	Transaction status. Available values: new , processing , completed , rejected , processing_realization_waiting , processing_realization Detailed description of operation statuses are presented in chapter <i>ATTACHMENT II - OPERATION STATUSES DESCRIPTION</i> <hr/> <b>Note:</b> completed and rejected status is final and will not change in the future. <hr/> Example: operation_status = completed
operation_amount	Transaction amount. The centesimal separator is a dot (.) typ: <i>string</i> min length: 1 max length: 10 Example: operation_amount = 177.27

Continued on next page

Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
operation_currency	The currency in which the amount parameter is given. Format according to ISO 4217 <sup>11</sup> 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
is_completed	Optional parameter indicating the transaction marked as completed in the seller's panel. Example: is_completed = false is_completed = true
operation_original_amount	The amount taken from amount parameter which has been sent by merchant's system. typ: <i>string</i> min length: 1 max length: 10 Example: operation_original_amount = 42.82 <hr/> <b>Important:</b> No verification of a operation_original_amount parameter on a merchant's site is unsecure and may cause financial loss. <hr/>
operation_original_currency	The currency taken from currency parameter which has been sent by merchant's system in redirecting buyer to the Dotpay website. Format compliant with standard ISO 4217 <sup>12</sup> . Example: operation_original_currency = EUR <hr/> <b>Important:</b> No verification of a operation_original_currency parameter on a merchant's site is unsecure and may cause financial loss. <hr/>
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

Continued on next page

Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
operation_related_number	<p>Number of related transaction, if any exists (e.g. in case of a refund).</p> <p>Format according to regular expression:</p> <p><code>^M\d{4,5}\-\d{4,5}\$</code></p> <p>Example:  <code>operation_related_number = M1234-56789</code></p> <p>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.</p>
control	<p>This value corresponds with <code>control</code> parameter which has been sent by merchant's service in a redirection to Dotpay service while making the payment (see <i>Table 2. (Additional parameters)</i> ).</p> <p>typ: <i>string</i>  max length: 1000</p> <p>Example:  <code>control = ec4bf09d3dbe0cb71e6abc3ea44a7273</code></p>
description	<p>This value corresponds with <code>description</code> parameter which has been sent by merchant's service in a redirection to Dotpay service while making a payment (see <i>Table 2. (Additional parameters)</i> ).</p> <p>typ: <i>string</i>  min length: 1  max length: 255</p> <p>Example:  <code>description = Invoice 20/2014</code></p>
email	<p>E-mail address of a person making the payment.</p> <p>typ: <i>string</i>  max length: 100</p> <p>Example:  <code>email = johnsmith@example.com</code></p>
p_info	<p>Name of a payment receiver which was displayed on the Dotpay payment website during Customer payment making.</p> <p>typ: <i>string</i>  max length: 300</p> <p>Example:  <code>p_info = Shop</code></p>
p_email	<p>E-mail address which was displayed for customer as contact to Seller.</p> <p>typ: <i>string</i>  max length: 100</p> <p>Example:  <code>p_email = contact@shop.com</code></p>

Continued on next page

Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
credit_card_issuer_identification_number	<p>Optional parameter. Masked payment card number with which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_issuer_identification_number = 603753</p>
credit_card_masked_number	<p>Optional parameter. Masked payment card number with which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_masked_number = XXXX XXXX XXXX 6214</p>
credit_card_expiration_year	<p>Optional parameter. Year expiration date of a payment card, which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_expiration_year = 2019</p>
credit_card_expiration_month	<p>Optional parameter. Month expiration date of a payment card, which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_expiration_month = 01</p>

Continued on next page

Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
credit_card_brand_codename	<p>Optional parameter. Payment card brand with which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_brand_codename = visa</p>
credit_card_brand_code	<p>Optional parameter. Payment card brand code with which payment has been made.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_brand_code = Visa</p>
credit_card_unique_identifier	<p>Optional parameter. The unique identifier of the card registered in Dotpay.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled.</p> <hr/> <p>Example: credit_card_unique_identifier = 9b73chvjxofy4d9g...d78d7153ju34po12</p>
credit_card_id	<p>Optional parameter. Payment card ID given by Dotpay system.</p> <hr/> <p><b>Important:</b> In order for parameter to be sent, options <i>HTTPS verify</i> and <i>SSL certificate verify</i> (menu <i>Settings -&gt; Notifications -&gt; Urlc configuration -&gt; Edit</i>) have to be enabled. Additionally given account ( id ) has to be configured on Dotpay side so <i>CHK</i> validation is required.</p> <hr/> <p>Example: credit_card_id = 59f92e2bf8bedc36...1369dvpa4b7ab02ae</p>
channel	<p>Payment channel (method) used by the buyer. For available values see <i>ATTACHMENT 1 - PAYMENT CHANNELS</i> .</p> <p>Example: channel = 1</p>

Continued on next page

Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
channel_country	Optional parameter. Indicates the country of the payment instrument from which payment was made. Format: ISO 3166-1 <sup>13</sup> (alpha-3). Example: channel_country = POL
geoup_country	Optional parameter. Indicates the location of the country resulting from IP address from which the payment was made. Format: ISO 3166-1 <sup>14</sup> (alpha-3). Example: geoup_country = POL
payer_bank_account_name	Optional parameter. Account owner name registered in the bank. Example: payer_bank_account_name = JAN KOWALSKI
payer_bank_account	Optional parameter. The bank account number from which the payment was made. Format compliant with the IBAN <sup>15</sup> . Example: payer_bank_account = PL41 1050 1009 4448 5481 1411 1395
payer_transfer_title	Optional parameter. The original title of the operation registered in the payer's bank. Example: payer_transfer_title = Store Name M5223-2008 Operation title Transaction No.: M52232008000000000000000000000000
blik_voucher_pin	Optional parameter. PIN for the issued BLIK vouchers. Example: blik_voucher_pin = 6365
blik_voucher_amount	Optional parameter. The nominal value of issued voucher BLIK. Example: blik_voucher_amount = 100.00
blik_voucher_amount_used	Optional parameter. Actual amount paid from the issued BLIK voucher. This amount may be equal to or less than the face value of the blik_voucher_amount voucher issued. Example: blik_voucher_amount_used = 60.00

Continued on next page





Tablica 3 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
signature	<p>The checksum created by an acronym function <i>SHA-256</i> being a concatenation of the above mentioned parameters according to the following formula:</p> $  \begin{aligned}  & PIN + id + operation\_number + operation\_type \\  & + operation\_status + operation\_amount + \\  & operation\_currency + operation\_withdrawal\_amount \\  & + operation\_commission\_amount + \\  & is\_completed + operation\_original\_amount + \\  & operation\_original\_currency + operation\_datetime \\  & + operation\_related\_number + control + \\  & description + email + p\_info + p\_email \\  & + credit\_card\_issuer\_identification\_number \\  & + credit\_card\_masked\_number + \\  & credit\_card\_expiration\_year + \\  & credit\_card\_expiration\_month + \\  & credit\_card\_brand\_codename + credit\_card\_brand\_code \\  & + credit\_card\_unique\_identifier + credit\_card\_id \\  & + channel + channel\_country + geoup\_country + \\  & payer\_bank\_account\_name + payer\_bank\_account \\  & + payer\_transfer\_title + blik\_voucher\_pin + \\  & blik\_voucher\_amount + blik\_voucher\_amount\_used + \\  & channel\_reference\_id + operation\_seller\_code  \end{aligned}  $ <hr/> <p><b>Important:</b> Sign + in the above formula has been used only for its readability. It is NOT used to calculate the checksum. If in sent notification some parameter is NOT present, should be considered as null value or omitted in the calculation.</p> <hr/> <p><b>Note:</b> <i>PIN</i> used to calculate a signature parameter is a string of characters which should be generated by a Merchant ( <i>id</i> )in My Dotpay panel (menu <i>Settings</i> -&gt; <i>Notifications</i> -&gt; <i>Urlc configuration</i>).</p> <p>It is only used to create the checksum and should NOT be sent along other payment data!</p> <hr/> <p><b>Important:</b> No verification of a signature parameter on a merchant's site is unsecure and may cause financial loss.</p>

### 3.2 An example of calculating the signature parameter

An example of calculating parameter signature value is presented below in PHP.

1  
2

```
<?php
```

(continues on next page)

<sup>11</sup> [https://en.wikipedia.org/wiki/ISO\\_4217](https://en.wikipedia.org/wiki/ISO_4217)

<sup>12</sup> [https://en.wikipedia.org/wiki/ISO\\_4217](https://en.wikipedia.org/wiki/ISO_4217)

<sup>13</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1](https://en.wikipedia.org/wiki/ISO_3166-1)

<sup>14</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1](https://en.wikipedia.org/wiki/ISO_3166-1)

<sup>15</sup> [https://en.wikipedia.org/wiki/International\\_Bank\\_Account\\_Number](https://en.wikipedia.org/wiki/International_Bank_Account_Number)

```

3 $PIN = "Np3n4QmXxp6M0TrLCVs905fdrGf3QIGm";
4
5 $sign =
6     $PIN,
7     $_POST['id'],
8     $_POST['operation_number'],
9     $_POST['operation_type'],
10    $_POST['operation_status'],
11    $_POST['operation_amount'],
12    $_POST['operation_currency'],
13    $_POST['operation_withdrawal_amount'],
14    $_POST['operation_commission_amount'],
15    $_POST['is_completed'],
16    $_POST['operation_original_amount'],
17    $_POST['operation_original_currency'],
18    $_POST['operation_datetime'],
19    $_POST['operation_related_number'],
20    $_POST['control'],
21    $_POST['description'],
22    $_POST['email'],
23    $_POST['p_info'],
24    $_POST['p_email'],
25    $_POST['credit_card_issuer_identification_number'],
26    $_POST['credit_card_masked_number'],
27    $_POST['credit_card_expiration_year'],
28    $_POST['credit_card_expiration_month'],
29    $_POST['credit_card_brand_codename'],
30    $_POST['credit_card_brand_code'],
31    $_POST['credit_card_unique_identifier'],
32    $_POST['credit_card_id'],
33    $_POST['channel'],
34    $_POST['channel_country'],
35    $_POST['geoip_country'],
36    $_POST['payer_bank_account_name'],
37    $_POST['payer_bank_account'],
38    $_POST['payer_transfer_title'],
39    $_POST['blik_voucher_pin'],
40    $_POST['blik_voucher_amount'],
41    $_POST['blik_voucher_amount_used'],
42    $_POST['channel_reference_id'],
43    $_POST['operation_seller_code'];
44
45 $signature=hash('sha256', $sign);
46
47 ?>

```

## 4 ADDITIONAL FEATURES

### 4.1 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 keeping the order.

```
PIN + api_version + lang + id + pid + 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 + deladdr  
+ 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 + application + application_version + 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 + surcharge_amount +  
surcharge + ignore_last_payment_channel + vco_call_id + vco_update_order_info  
+ vco_subtotal + vco_shipping_handling + vco_tax + vco_discount + vco_gift_wrap  
+ vco_misc + vco_promo_code + credit_card_security_code_required +  
credit_card_operation_type + credit_card_avs + credit_card_threeds + customer  
+ gp_token + blik_refusenopayid + auto_reject_date + ap_token + id1 + amount1  
+ currency1 + description1 + controll + ... + id(n) + amount(n) + currency(n) +  
description(n) + control(n)
```

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

In case of links generated using seller administration panel ( `pid` ), calculation of `chk` value is done on basis of `PIN + pid` (according to the above formula), don't take into account the parameters from which `pid` was created.

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

---

**Note:** `PIN` used to calculate a `chk` parameter is a string of characters which should be generated by a Merchant ( `id` ) in My Dotpay panel (menu *Settings* -> *Notifications* -> *Urlc configuration*).

---

**Caution:** Default shop ( `id` ) configuration verifies `chk` parameter sent in payment request.

If it's absent or its value is incorrect, error will be returned.

`chk` verification isn't the only payment security mechanism, in some cases it might prove to be unnecessary. `chk` verification allows to check parameters in the first step of payment: redirecting customer to Dotpay. The most important part is compare amount and currency from payment created in Dotpay with data saved in seller's service - *RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)*.

If your shop doesn't have `chk` verification enabled (it's optional or completely disabled), ask Dotpay to change your account's settings.

If you are not sure your integration supports `chk`, please contact your service provider / programmer responsible for the implementation, or Doptay Tech support (email: [tech@dotpay.pl](mailto:tech@dotpay.pl) ).

In case you don't want to use this security measure and are aware of possible implications, you can disable `chk` for your account ( `id` ). To do so, send an email to [33](mailto:admini-</a></p></div><div data-bbox=)

stracja@dotpay.pl

Below are exemplary functions (PHP) generating *POST* / *GET* request with *chk* .

#### 4.1.1 Basic Example

Download this example (change extension to *.php*): [Download example](#)

```
1 <?php
2
3 ##### https://www.dotpay.pl/developer/doc/api_payment/ #####
4 ↳#####
5 #
6 #     Exemplary function (PHP) generating the correct payment,
7 ↳redirection (POST / GET) to Dotpay payment api with parametr 'chk'
8 ↳(checksum).
9 #     You enter the payment data in the parameter: $ParametersArray.
10 #
11 #
12 #     Dotpay Sp. z o.o.
13 #     Tech Customer Service: tech@dotpay.pl
14 #     Date: 2020-02-11
15 #
16 #####
17 ↳#####
18
19 /** ----- BASE CONFIG ----- */
20
21 // Your Dotpay ID shop (6 digits)
22 $DotpayId = "123456";
23
24 // PIN for Your Dotpay ID (copy this from your dotpay panel carefully,
25 ↳without space)
26 $DotpayPin = "MyDotpayPIN000000j4suuSgMPXlg100";
27
28 // Dotpay Environment, available: "test" or "production"
29 $Environment = "test";
30
31 //Redirection method: POST or GET ; recommended method is "POST"
32 $RedirectionMethod = "POST";
33
34 /** ----- end config ----- */
35
36 // ** ----- SAMPLE DATA ----- **/
37
38 /* ## SAMPLE PAYMENT DATA IN ## */
39 // Note! You can use more parameters if You need
40 // You must give at least: 'amount', 'currency', 'description'
41 ↳(and of course ID and PIN in the configuration of this script)
42 // see more: https://www.dotpay.pl/developer/doc/api_payment/en/
43 ↳index.html#tabelka-1-podstawowe-parametry-przesylane-do-serwisu-dotpay
44 // and: https://www.dotpay.pl/developer/doc/api_payment/en/index.
45 ↳html#tabelka-2-dodatkowe-parametry-przesylane-do-serwisu-dotpay
```

(continues on next page)

```

41 // -----
42 $ParametersArray = array(
43     "api_version" => "dev",
44     "amount" => "100.00",
45     "currency" => "PLN",
46     "description" => "Order no. 567915976",
47     "url" => "https://www.example.com/thanks_page.php",
48     "type" => "0",
49     "buttontext" => "Back to the www.example.com",
50     "urlc" => "https://www.example.com/urlc_receiver.php",
51     "control" => "M1231MzaUdLQWR3",
52     "firstname" => "Jan",
53     "lastname" => "Nowak",
54     "email" => "jan.nowak@example.com",
55     "street" => "Warszawska",
56     "street_n1" => "1",
57     "city" => "Krakow",
58     "postcode" => "12-345",
59     "phone" => "123456789",
60     "country" => "POL",
61     "ignore_last_payment_channel" => 1
62 );
63
64 // ** ----- SAMPLE DATA end -----
65 →**/
66
67
68
69 ## CALCULATE CHECKSUM - CHK
70
71 function GenerateChk($DotpayId, $DotpayPin, $ParametersArray)
72 {
73     $ParametersArray['id'] = $DotpayId;
74     $CHKInputString = $DotpayPin.
75         (isset($ParametersArray['api_version']) ?
76 →$ParametersArray['api_version'] : null).
77         (isset($ParametersArray['lang']) ?
78 →$ParametersArray['lang'] : null).
79         (isset($ParametersArray['id']) ? $ParametersArray[
80 →'id'] : null).
81         (isset($ParametersArray['amount']) ?
82 →$ParametersArray['amount'] : null).
83         (isset($ParametersArray['currency']) ?
84 →$ParametersArray['currency'] : null).
85         (isset($ParametersArray['description']) ?
86 →$ParametersArray['description'] : null).
87         (isset($ParametersArray['control']) ?
88 →$ParametersArray['control'] : null).
89         (isset($ParametersArray['channel']) ?
90 →$ParametersArray['channel'] : null).
91         (isset($ParametersArray['url']) ? $ParametersArray[
92 →'url'] : null).

```

(continues on next page)

(continued from previous page)

```
87         (isset($ParametersArray['type']) ?
↳$ParametersArray['type'] : null).
88         (isset($ParametersArray['buttontext']) ?
↳$ParametersArray['buttontext'] : null).
89         (isset($ParametersArray['urlc']) ?
↳$ParametersArray['urlc'] : null).
90         (isset($ParametersArray['firstname']) ?
↳$ParametersArray['firstname'] : null).
91         (isset($ParametersArray['lastname']) ?
↳$ParametersArray['lastname'] : null).
92         (isset($ParametersArray['email']) ?
↳$ParametersArray['email'] : null).
93         (isset($ParametersArray['street']) ?
↳$ParametersArray['street'] : null).
94         (isset($ParametersArray['street_n1']) ?
↳$ParametersArray['street_n1'] : null).
95         (isset($ParametersArray['city']) ?
↳$ParametersArray['city'] : null).
96         (isset($ParametersArray['postcode']) ?
↳$ParametersArray['postcode'] : null).
97         (isset($ParametersArray['phone']) ?
↳$ParametersArray['phone'] : null).
98         (isset($ParametersArray['country']) ?
↳$ParametersArray['country'] : null).
99         (isset($ParametersArray['ignore_last_payment_
↳channel']) ? $ParametersArray['ignore_last_payment_channel'] : null);
↳
100
101     return hash('sha256', $CHKInputString);
102 }
103
104
105
106 ## GENERATE FORM TO DOTPAY
107
108 function GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
↳$RedirectionMethod, $ParametersArray)
109 {
110     $ChkValue = GenerateChk($DotpayId, $DotpayPin, $ParametersArray);
111
112     if ($Environment == 'production') {
113         $EnvironmentAddress = 'https://ssl.dotpay.pl/t2/';
114     } elseif ($Environment == 'test') {
115         $EnvironmentAddress = 'https://ssl.dotpay.pl/test_payment/';
116     }
117
118     if ($RedirectionMethod == 'POST') {
119         $RedirectionCode = '<form action="'. $EnvironmentAddress. '" method=
↳"POST" id="dotpay_redirection_form" accept-charset="UTF-8">'.PHP_EOL;
120         $RedirectionCode .= "\t".<input name="id" value="'.
↳$DotpayId.'" type="hidden"/>'.PHP_EOL;
121
122         foreach ($ParametersArray as $key => $value)
123         {
124             $RedirectionCode .= "\t".<input name="'. $key. '" value="'.
↳$value.'" type="hidden"/>'.PHP_EOL;
```

(continues on next page)

(continued from previous page)

```
125     }
126     $RedirectionCode .= "\t".<input name="chk" value="' . $ChkValue . "' <
↳ type="hidden"/>'.PHP_EOL;
127     $RedirectionCode .= '</form>'.PHP_EOL.<button id="dotpay_
↳ redirection_button" type="submit" form="dotpay_redirection_form" value=
↳ "Submit">Confirm and Pay</button>'.PHP_EOL;
128
129     return $RedirectionCode;
130
131 } elseif ($RedirectionMethod == 'GET') {
132     $RedirectionCode = $EnvironmentAddress.'?';
133
134     foreach ($ParametersArray as $key => $value)
135     {
136         $RedirectionCode .= $key.'=' .rawurlencode($value).'&';
137     }
138
139     $RedirectionCode .= 'id='.$DotpayId;
140     $RedirectionCode .= '&chk='.$ChkValue;
141
142     return '<a href="' . $RedirectionCode . '">Link to Pay</a>';
143 }
144 }
145
146
147 ## get form (POST method) or payment link (GET method)
148 ## ("Dotpay ID", "PIN", "[test|production]", "[POST|GET]", "payment data")
149
150 echo GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
↳ $RedirectionMethod , $ParametersArray);
151
152 ?>
```

#### 4.1.2 Full Example

Download this example (change extension to .php): [Download example](#)

```
1 <?php
2
3 ##### https://www.dotpay.pl/developer/doc/api_payment/ #####
↳ #####
4 #
5 #     Exemplary function (PHP) generating the correct payment_
↳ redirection (POST / GET) to Dotpay payment api with parametr 'chk'_
↳ (checksum).
6 #     You enter the payment data in the parameter: $ParametersArray.
7 #
8 #     In addition, the exemplary includes the use of data for the_
↳ Multimerchant service ($MultiMerchantList)
9 #     and some payment channels that require additional data, e.g. a_
↳ delivery address ($customer).
10 #     You do not need to use them if you do not use these features.
11 #
12 #     Dotpay Sp. z o.o.
13 #     Tech Customer Service: tech@dotpay.pl
```

(continues on next page)



```

14 #   Date: 2019-04-15
15 #
16 #####
17 ↪#####
18
19 /** -----  BASE CONFIG  -----  **/
20
21 // Your Dotpay ID shop (6 digits)
22     $DotpayId = "123456";
23
24 // PIN for Your Dotpay ID (copy this from your dotpay panel carefully,
25 ↪without space)
26     $DotpayPin = "0i9xSl8lE4cJhygjKn2LlMW2MBqkv234";
27
28 // Dotpay Environment, available: "test" or "production"
29     $Environment = "test";
30
31 //Redirection method: POST or GET ; recommended method is "POST"
32     $RedirectionMethod = "POST";
33
34 /** -----  end config  -----  **/
35
36 // Do not remove this!
37 $MultiMerchantList = array(); //optional data
38 $customer = null; //optional data
39
40
41 ## CALCULATE CHECKSUM - CHK
42
43 function GenerateChk($DotpayId, $DotpayPin, $Environment,
44 ↪$RedirectionMethod, $ParametersArray, $MultiMerchantList, $customer_
45 ↪base64)
46
47 {
48     $ParametersArray['id'] = $DotpayId;
49     $ParametersArray['customer'] = $customer_base64;
50
51     $chk = $DotpayPin.
52     (isset($ParametersArray['api_version']) ? $ParametersArray['api_version
53 ↪'] : null).
54     (isset($ParametersArray['lang']) ? $ParametersArray['lang'] : null).
55     (isset($ParametersArray['id']) ? $ParametersArray['id'] : null).
56     (isset($ParametersArray['pid']) ? $ParametersArray['pid'] : null).
57     (isset($ParametersArray['amount']) ? $ParametersArray['amount'] :
58 ↪null).
59     (isset($ParametersArray['currency']) ? $ParametersArray['currency'] :
60 ↪null).
61     (isset($ParametersArray['description']) ? $ParametersArray['description
62 ↪'] : null).
63     (isset($ParametersArray['control']) ? $ParametersArray['control'] :
64 ↪null).
65     (isset($ParametersArray['channel']) ? $ParametersArray['channel'] :
66 ↪null).
67     (isset($ParametersArray['credit_card_brand']) ? $ParametersArray[
68 ↪'credit_card_brand'] : null).

```

(continues on next page)

```

60     (isset($ParametersArray['ch_lock']) ? $ParametersArray['ch_lock'] :
↳null).
61     (isset($ParametersArray['channel_groups']) ? $ParametersArray['channel_
↳groups'] : null).
62     (isset($ParametersArray['onlinetransfer']) ? $ParametersArray[
↳'onlinetransfer'] : null).
63     (isset($ParametersArray['url']) ? $ParametersArray['url'] : null).
64     (isset($ParametersArray['type']) ? $ParametersArray['type'] : null).
65     (isset($ParametersArray['buttontext']) ? $ParametersArray['buttontext
↳'] : null).
66     (isset($ParametersArray['urllc']) ? $ParametersArray['urllc'] : null).
67     (isset($ParametersArray['firstname']) ? $ParametersArray['firstname']
↳: null).
68     (isset($ParametersArray['lastname']) ? $ParametersArray['lastname'] :
↳null).
69     (isset($ParametersArray['email']) ? $ParametersArray['email'] : null).
70     (isset($ParametersArray['street']) ? $ParametersArray['street'] :
↳null).
71     (isset($ParametersArray['street_n1']) ? $ParametersArray['street_n1']
↳: null).
72     (isset($ParametersArray['street_n2']) ? $ParametersArray['street_n2']
↳: null).
73     (isset($ParametersArray['state']) ? $ParametersArray['state'] : null).
74     (isset($ParametersArray['addr3']) ? $ParametersArray['addr3'] : null).
75     (isset($ParametersArray['city']) ? $ParametersArray['city'] : null).
76     (isset($ParametersArray['postcode']) ? $ParametersArray['postcode'] :
↳null).
77     (isset($ParametersArray['phone']) ? $ParametersArray['phone'] : null).
78     (isset($ParametersArray['country']) ? $ParametersArray['country'] :
↳null).
79     (isset($ParametersArray['code']) ? $ParametersArray['code'] : null).
80     (isset($ParametersArray['p_info']) ? $ParametersArray['p_info'] :
↳null).
81     (isset($ParametersArray['p_email']) ? $ParametersArray['p_email'] :
↳null).
82     (isset($ParametersArray['n_email']) ? $ParametersArray['n_email'] :
↳null).
83     (isset($ParametersArray['expiration_date']) ? $ParametersArray[
↳'expiration_date'] : null).
84     (isset($ParametersArray['deladdr']) ? $ParametersArray['deladdr'] :
↳null).
85     (isset($ParametersArray['recipient_account_number']) ?
↳$ParametersArray['recipient_account_number'] : null).
86     (isset($ParametersArray['recipient_company']) ? $ParametersArray[
↳'recipient_company'] : null).
87     (isset($ParametersArray['recipient_first_name']) ? $ParametersArray[
↳'recipient_first_name'] : null).
88     (isset($ParametersArray['recipient_last_name']) ? $ParametersArray[
↳'recipient_last_name'] : null).
89     (isset($ParametersArray['recipient_address_street']) ?
↳$ParametersArray['recipient_address_street'] : null).
90     (isset($ParametersArray['recipient_address_building']) ?
↳$ParametersArray['recipient_address_building'] : null).
91     (isset($ParametersArray['recipient_address_apartment']) ?
↳$ParametersArray['recipient_address_apartment'] : null).
92     (isset($ParametersArray['recipient_address_postcode']) ?
↳$ParametersArray['recipient_address_postcode'] : null). (continues on next page)

```

```

93     (isset($ParametersArray['recipient_address_city']) ? $ParametersArray[
↪'recipient_address_city'] : null).
94     (isset($ParametersArray['application']) ? $ParametersArray['application
↪'] : null).
95     (isset($ParametersArray['application_version']) ? $ParametersArray[
↪'application_version'] : null).
96     (isset($ParametersArray['warranty']) ? $ParametersArray['warranty'] :
↪null).
97     (isset($ParametersArray['bylaw']) ? $ParametersArray['bylaw'] : null).
98     (isset($ParametersArray['personal_data']) ? $ParametersArray['personal_
↪data'] : null).
99     (isset($ParametersArray['credit_card_number']) ? $ParametersArray[
↪'credit_card_number'] : null).
100    (isset($ParametersArray['credit_card_expiration_date_year']) ?
↪$ParametersArray['credit_card_expiration_date_year'] : null).
101    (isset($ParametersArray['credit_card_expiration_date_month']) ?
↪$ParametersArray['credit_card_expiration_date_month'] : null).
102    (isset($ParametersArray['credit_card_security_code']) ?
↪$ParametersArray['credit_card_security_code'] : null).
103    (isset($ParametersArray['credit_card_store']) ? $ParametersArray[
↪'credit_card_store'] : null).
104    (isset($ParametersArray['credit_card_store_security_code']) ?
↪$ParametersArray['credit_card_store_security_code'] : null).
105    (isset($ParametersArray['credit_card_customer_id']) ? $ParametersArray[
↪'credit_card_customer_id'] : null).
106    (isset($ParametersArray['credit_card_id']) ? $ParametersArray['credit_
↪card_id'] : null).
107    (isset($ParametersArray['blik_code']) ? $ParametersArray['blik_code']
↪: null).
108    (isset($ParametersArray['credit_card_registration']) ?
↪$ParametersArray['credit_card_registration'] : null).
109    (isset($ParametersArray['surcharge_amount']) ? $ParametersArray[
↪'surcharge_amount'] : null).
110    (isset($ParametersArray['surcharge']) ? $ParametersArray['surcharge']
↪: null).
111    (isset($ParametersArray['surcharge']) ? $ParametersArray['surcharge']
↪: null).
112    (isset($ParametersArray['ignore_last_payment_channel']) ?
↪$ParametersArray['ignore_last_payment_channel'] : null).
113    (isset($ParametersArray['vco_call_id']) ? $ParametersArray['vco_call_id
↪'] : null).
114    (isset($ParametersArray['vco_update_order_info']) ? $ParametersArray[
↪'vco_update_order_info'] : null).
115    (isset($ParametersArray['vco_subtotal']) ? $ParametersArray['vco_
↪subtotal'] : null).
116    (isset($ParametersArray['vco_shipping_handling']) ? $ParametersArray[
↪'vco_shipping_handling'] : null).
117    (isset($ParametersArray['vco_tax']) ? $ParametersArray['vco_tax'] :
↪null).
118    (isset($ParametersArray['vco_discount']) ? $ParametersArray['vco_
↪discount'] : null).
119    (isset($ParametersArray['vco_gift_wrap']) ? $ParametersArray['vco_gift_
↪wrap'] : null).
120    (isset($ParametersArray['vco_misc']) ? $ParametersArray['vco_misc'] :
↪null).
121    (isset($ParametersArray['vco_promo_code']) ? $ParametersArray['vco_
↪promo_code'] : null).

```

(continues on next page)

(continued from previous page)

```
122     (isset($ParametersArray['credit_card_security_code_required']) ?  
123     ↪$ParametersArray['credit_card_security_code_required'] : null).  
124     (isset($ParametersArray['credit_card_operation_type']) ?  
125     ↪$ParametersArray['credit_card_operation_type'] : null).  
126     (isset($ParametersArray['credit_card_avs']) ? $ParametersArray['credit_  
127     ↪card_avs'] : null).  
128     (isset($ParametersArray['credit_card_threeds']) ? $ParametersArray[  
129     ↪'credit_card_threeds'] : null).  
130     (isset($ParametersArray['customer']) ? $ParametersArray['customer'] :  
131     ↪null).  
132     (isset($ParametersArray['gp_token']) ? $ParametersArray['gp_token'] :  
133     ↪null).  
134     (isset($ParametersArray['blik_refusenopayid']) ? $ParametersArray[  
135     ↪'blik_refusenopayid'] : null).  
136     (isset($ParametersArray['auto_reject_date']) ? $ParametersArray['auto_  
137     ↪reject_date'] : null).  
138     (isset($ParametersArray['ap_token']) ? $ParametersArray['ap_token'] :  
139     ↪null);  
140  
141     foreach ($MultiMerchantList as $item) {  
142         foreach ($item as $key => $value) {  
143             $chk = $chk.  
144             (isset($value) ? $value : null);  
145         }  
146     }  
147     return $chk;  
148 }  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

(continues on next page)

```

164     }
165
166         if(isset($customer_base64)) {
167             $RedirectionCode .= "\t".<input name="customer"
↪value="'. $customer_base64.'" type="hidden"/>'.PHP_EOL;
168         }
169
170         foreach ($MultiMerchantList as $item) {
171             foreach ($item as $key => $value) {
172                 $RedirectionCode .= "\t".<input name="'. $key.'" value="'.
↪$value.'" type="hidden"/>'.PHP_EOL;
173             }
174         }
175
176         $RedirectionCode .= "\t".<input name="chk" value="'. $ChkValue.'"
↪type="hidden"/>'.PHP_EOL;
177         $RedirectionCode .= '</form>'.PHP_EOL.<button id="dotpay_
↪redirection_button" type="submit" form="dotpay_redirection_form" value=
↪"Submit">Confirm and Pay</button>'.PHP_EOL;
178
179         return $RedirectionCode;
180
181     } elseif ($RedirectionMethod == 'GET') {
182         $RedirectionCode = $EnvironmentAddress.'?';
183
184         foreach ($ParametersArray as $key => $value) {
185             $RedirectionCode .= $key.'='.rawurlencode($value).'&';
186         }
187
188         if(isset($customer_base64)) {
189             $RedirectionCode .= 'customer='.$customer_base64.'&
↪';
190         }
191
192         foreach ($MultiMerchantList as $item) {
193             foreach ($item as $key => $value) {
194                 $RedirectionCode .= $key.'='.rawurlencode($value).'&';
195             }
196         }
197
198         $RedirectionCode .= 'chk='.$ChkValue;
199
200         return '<a href="'. $RedirectionCode.'">Go to Pay</a><br>
↪link:<br>'. $RedirectionCode;
201     }
202 }
203
204
205
206
207
208 // ** ----- SAMPLE DATA ----- **/
209
210
211
212 /* ## SAMPLE PAYMENT DATA IN ## */

```

(continues on next page)

(continued from previous page)

```
213 // Note! You can use more parameters if You need
214 // You must give at least: 'amount', 'currency', 'description'
↳(and of course ID and PIN in the configuration of this script)
215 // see more: https://www.dotpay.pl/developer/doc/api_payment/en/
↳index.html#tabela-1-podstawowe-parametry-przesylane-do-serwisu-dotpay
216 // and: https://www.dotpay.pl/developer/doc/api_payment/en/index.
↳html#tabela-2-dodatkowe-parametry-przesylane-do-serwisu-dotpay
217
218 // -----
219 $ParametersArray = array(
220     "api_version" => "dev",
221     "amount" => "100.00",
222     "currency" => "PLN",
223     "description" => "Order no. 567915976",
224     "url" => "https://www.example.com/thanks_page.php",
225     "type" => "4",
226     "channel" => "2",
227     "buttontext" => "Back to the www.example.com",
228     "urlc" => "https://www.example.com/urlc_receiver.php",
229     "control" => "M1231MzaUdLQWR3",
230     "firstname" => "Jan",
231     "lastname" => "Nowak",
232     "email" => "jan.nowak@example.com",
233     "street" => "Warszawska",
234     "street_n1" => "1",
235     "city" => "Krakow",
236     "postcode" => "12-345",
237     "phone" => "123456789",
238     "country" => "POL",
239     "ignore_last_payment_channel" => "true"
240 );
241 // -----
242
243
244 /*   ###  SAMPLE CUSTOMER DATA IN with delivery address (optional)   ###
245         You can remove it if You don't need it
246 */
247
248 // -----
249 $customer = array(
250     "payer" => array(
251         "first_name" => "Jan",
252         "last_name" => "Kowal",
253         "email" => "mymail@example.com",
254     ) ,
255     "order" => array(
256         "delivery_address" => array(
257             "city" => "Kraków",
258             "street" => "Wielicka",
259             "building_number" => "28B",
260             "postcode" => "30-552",
261         )
262     )
263 );
264
265 // -----
```

(continues on next page)

```

266
267
268
269 /* ### SAMPLE Multimerchant DATA IN (3 accounts 'child' type ) - optional_
    ↳ ###
270         You can remove it if You don't need it
271         see more: https://www.dotpay.pl/developer/doc/api_payment/
    ↳ en/index.html#platnosc-dzielona-multimerchant-pasaz
272 */
273
274 // ----- uncomment if you need:
275
276 /*
277     $MultiMerchantList = array(
278         $MultiMerchant1 = array(
279             "id1" => "123456",
280             "amount1" => "10.00",
281             "currency1" => "PLN",
282             "description1" => "description1",
283             "control1" => "control1",
284         ),
285         $MultiMerchant2 = array(
286             "id2" => "234561",
287             "amount2" => "60.00",
288             "currency2" => "PLN",
289             "description2" => "description2",
290             "control2" => "control2",
291         ),
292         $MultiMerchant3 = array(
293             "id3" => "234562",
294             "amount3" => "30.00",
295             "currency3" => "PLN",
296             "description3" => "description3",
297             "control3" => "control3",
298         )
299     );
300
301 */
302 // -----
303
304 // ** ----- SAMPLE DATA end -----
    ↳ **/
305
306
307
308 if (empty($customer) || !isset($customer['payer']) || !isset($customer[
    ↳ 'order']['delivery_address'])) {
309     $customer_base64 = null;
310 } else {
311     $customer_base64 = base64_encode(json_encode($customer));
312 }
313
314
315
316 ## get form (POST method) or payment link (GET method)
317 ## ("Dotpay ID", "PIN", "[test|production]", "[POST|GET]", "payment data",
    ↳ "Multimerchant data", "additional customer data")

```

(continues on next page)

```

318
319 echo GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
    ↳$RedirectionMethod , $ParametersArray, $MultiMerchantList, $customer_
    ↳base64);
320
321 ?>

```

## 4.2 One-click and recurring payments

---

**Note:** Functionality is only available for shops ( id ) which were adequately configured by Dotpay, what is dependent on signed agreement.

---

Functionality is only available for payment card channel (number 248), allowing client to realize payments without entering complete card data in future transactions made via Dotpay.

In order to use the functionality seller's system has to send additional parameters (described below) in payment request and client should allow his card information to be stored by Dotpay (all data is stored according to highest security standards - PCI DSS Level 1).

When using described functionality it is recommended for request sent to Dotpay to be signed as well as signature verification to be required on given shop ( id ). Description of signature creation can be found in *Redirection parameters integrity protection (CHK)* section.

In order to register client's card data in Dotpay system additional parameters have to be sent:

### 4.2.1 Table 4. (Parameters for card registration)

PARAMETER	MEANING / DESCRIPTION
credit_card_store	Parameter defining whether client card data should be stored by Dotpay. Available values: 1 - Storing credit card data is accepted Example: credit_card_store = 1
credit_card_customer_id	Unique buyer ID generated and stored by seller's system. Required for future payments  <b>Important:</b> Dotpay does <b>NOT</b> return this information anywhere (i.e. <i>URLC</i> or <i>API</i> ).  type: <i>string</i> min length: 4 max length 1024 Example: credit_card_customer_id = f9c6a4-25473

Continued on next page



Tablica 4 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
credit_card_registration	<p>Optional parameter. Creates operation credit_card_registration instead of payment, what allows to register credit card without paying. Instead of charge temporary funds blockade will be issued, which is lifted when registration process is completed.</p> <hr/> <p><b>Tip:</b> For operation credit_card_registration it's required to use an amount of value 1.00 ( amount = 1.00).</p> <hr/> <p>Available values: true - registration without payment Example: credit_card_registration = true</p>

In order to make payment with registered card additional parameters described below need to be sent:

#### 4.2.2 Table 5. (Parameters for consecutive payments with previously registered card)

PARAMETER	MEANING / DESCRIPTION
credit_card_customer_id	Unique buyer ID generated by seller's system described in table above.
credit_card_id	Buyer's registered card ID This information might be sent in <i>URLC</i> notification (special account configuration required, described in <i>RECEIVING PAYMENT CONFIRMATION (URLC CONFIRMATION)</i> section). It also might be received via <a href="#">API<sup>16</sup></a> from operation details.

Common parameters for both first and consecutive payments have been described in table below, all of them are optional. Standard account configuration doesn't allow them to be used (default settings are applied, configurable by Dotpay), special agreement might be required in order to send them.

#### 4.2.3 Table 6. (Common parameters for both first and consecutive payments)

<sup>16</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/en/#single-operation-details](https://www.dotpay.pl/developer/doc/api_seller/en/#single-operation-details)

PARAMETER	MEANING / DESCRIPTION
credit_card_operation_type	<p>Optional parameter. Operation type. Available values: e_commerce - first and consecutive payment in one-click model (default value), recurring_init - first transaction allowing later use of recurring payments, recurring - recurring payment (customer doesn't have to be present in order to charge the registered card), Account ( id ) might be configured by Dotpay so recurring payments are default setting. In this case appropriate types recurring_init and recurring are going to be set automatically. Parameter is required when one account ( id ) handles both e-commerce and recurring payments.</p> <hr/> <p><b>Tip:</b> Card registered using e_commerce cannot be used for recurring operation. Cards registered using recurring_init can be used for both recurring and e_commerce .</p> <hr/>
credit_card_security_code_required	<p>Optional parameter. Allows to control whether CVV/CVV2 security code is required during payment. Applies only to consecutive e_commerce (one click). Available values: yes - required (default value), no - not required, option - field for security code will show up on payment form, but it won't be mandatory.</p>
credit_card_threeds	<p>Optional parameter. Allows to control whether 3-D Secure authentication code is required during payment. Applies only to e_commerce model for enrolled cards. Available values: yes - required (default value, recommended for transaction security), no - not required,</p> <hr/> <p><b>Tip:</b> Operation type recurring_init requires and recurring doesn't require 3-D Secure authentication code, ignoring account settings or sent parameters.</p> <hr/>
credit_card_avs	<p>Optional parameter. Allows to control whether customer will be required to enter additional personal data on payment form (address and phone number). Available values: yes - required, no - not required (default value).</p>

#### 4.2.4 Exemplary integration models and their corresponding requirements)

## Integration model: 1 click

Tablica7: Table 7: Requirements for integration type 1 Click

Card registration	Consecutive payment
Required parameters: - credit_card_customer_id - credit_card_store = 1 Payment process: - CVV - required - 3DS - required*	Required parameters: - credit_card_customer_id - credit_card_id Payment process: - CVV - optional - 3DS - required*

\* - recommended for transaction security

## Integration model: recurring

Tablica8: Table 8: Requirements for integration type recurring

Card registration	Consecutive payment
Required parameters: - credit_card_customer_id - credit_card_store = 1 Payment process: - CVV - required - 3DS - required	Required parameters: - credit_card_customer_id - credit_card_id Payment process: - CVV - not applicable - 3DS - not applicable

\* - recommended for transaction security

## Mixed integration model: 1 click + recurring

Tablica9: Table 9: Requirements for integration type 1 click + recurring . Default account setting: e-commerce

Integration model	Card registration	Consecutive payment
1 click	Required parameters: - credit_card_customer_id - credit_card_store = 1 Payment process: - CVV - required - 3DS - required*	Required parameters: - credit_card_customer_id - credit_card_id Payment process: - CVV - optional - 3DS - required*
recurring	Required parameters: - credit_card_customer_id - credit_card_store = 1 - credit_card_operation_type = recurring_init Payment process: - CVV - required - 3DS - required	Required parameters: - credit_card_customer_id - credit_card_id - credit_card_operation_type = recurring Payment process: - CVV - not applicable - 3DS - not applicable

\* - recommended for transaction security

Tablica10: Table 10: Requirements for integration type 1  
*click + recurring* Default account setting: recurring

Integration model	Card registration	Consecutive payment
1 click	Required parameters: - credit_card_customer_id - credit_card_store = 1 - credit_card_operation_type = e_commerce Payment process: - CVV - required - 3DS - required*	Required parameters: - credit_card_customer_id - credit_card_id - credit_card_operation_type = e_commerce Payment process: - CVV - optional - 3DS - required*
recurring	Required parameters: - credit_card_customer_id - credit_card_store = 1 Payment process: - CVV - required - 3DS - required	Required parameters: - credit_card_customer_id - credit_card_id Payment process: - CVV - not applicable - 3DS - not applicable

\* - recommended for transaction security

Below are exemplary redirection forms for one-click payment realization.

Exemplary payment request with card registration:

```

1 <div>
2
3   <form action="https://ssl.dotpay.pl/t2/" method="post" id="dotpay_
  ↳ redirection_form">
4     <input name="api_version" value="dev" type="hidden" />
5     <input name="id" value="123456" type="hidden" />
6     <input name="amount" value="320.00" type="hidden" />
7     <input name="currency" value="PLN" type="hidden" />
8     <input name="description" value="Płatność za 12345/2014" type="hidden"
  ↳ />
9     <input name="control" value="202cb962ac590" type="hidden" />
10    <input name="channel" value="248" type="hidden" />
11    <input name="ch_lock" value="1" type="hidden" />
12    <input name="firstname" value="John" type="hidden" />
13    <input name="lastname" value="Smith" type="hidden" />
14    <input name="email" value="john.smith@example.com" type="hidden" />
15    <input name="type" value="0" type="hidden" />
16    <input name="url" value="https://www.example.com/thanks_page.php" type=
  ↳ "hidden" />
17    <input name="urlc" value="https://www.example.com/urlc_receiver.php"
  ↳ type="hidden" />
18    <input name="credit_card_store" value="1" type="hidden" />
19    <input name="credit_card_customer_id" value="f9c6a4-25473" type="hidden
  ↳ " />
20    <input name="chk" value="11ac1938ac47ddd53815b4aeb6230ab9fe4554d82ee11e39c41b9055f3
  ↳ " type="hidden" />
21  </form>
22  <p>
23    <button type="submit" form="dotpay_redirection_form" value="Submit">
24      Potwierdź zamówienie i zapłać</button>
25  </p>
26
27 </div>

```

Exemplary payment request using registered card data (one-click):

```
1 <div>
2   <form action="https://ssl.dotpay.pl/t2/" method="post" id="dotpay_
  ↳ redirection_form">
3
4     <input name="api_version" value="dev" type="hidden" />
5     <input name="id" value="123456" type="hidden" />
6     <input name="amount" value="410.00" type="hidden" />
7     <input name="currency" value="PLN" type="hidden" />
8     <input name="description" value="Płatność za 12346/2014" type="hidden"
  ↳ />
9     <input name="control" value="31ee79b30dc39a9cbaa" type="hidden" />
10    <input name="channel" value="248" type="hidden" />
11    <input name="ch_lock" value="1" type="hidden" />
12    <input name="firstname" value="John" type="hidden" />
13    <input name="lastname" value="Smith" type="hidden" />
14    <input name="email" value="john.smith@example.com" type="hidden" />
15    <input name="type" value="4" type="hidden" />
16    <input name="url" value="https://www.example.com/thanks_page.php" type=
  ↳ "hidden" />
17    <input name="urlc" value="https://www.example.com/urlc_receiver.php"
  ↳ type="hidden" />
18    <input name="credit_card_customer_id" value="f9c6a4-25473" type="hidden
  ↳ " />
19    <input name="credit_card_id" value="59f92e2bf8bedc36bec2219862448dd54d.
  ↳ ..1829a239eb7432d0easuxp2w158eb13d6333ce71369184eb7ab02ae" type="hidden"
  ↳ />
20    <input name="chk" value="ed0ef4e488ec2de3135b0a1ca226c31867f78bbcd8fe06506ae666210a
  ↳ " type="hidden" />
21
22  </form>
23
24  <p>
25    <button type="submit" form="dotpay_redirection_form" value="Submit">
  ↳ Potwierdź zamówienie i zapłać (płatność one-click)</button>
26  </p>
27 </div>
```

---

**Note:** For recurring payments it is recommended to use direct communication with Dotpay using REST API ( register order ).

---

Table below describes parameters which allow payment card data to be sent to Dotpay system if it is entered by client on seller's website (for example on order summary page) instead of Dotpay.

---

**Note:** Processing payment card data by seller's system requires - according to Payment Card Industry Data Security Standard (PCI DSS) - additional conditions to be met.

In order to receive more information about necessary formalities please contact Sales Department ([handlowy@dotpay.pl](mailto:handlowy@dotpay.pl)).

---

**Caution:** Sending payment card data with parameters described below is only possible via *POST* method.

#### 4.2.5 Table 11. (Parameters for credit card data)

PARAMETER	MEANING / DESCRIPTION
credit_card_number	Client's payment card number. type: <i>credit_card_number</i> max length: 26 regular expression: <code>^\d\s{12,26}\$</code> Example: credit_card_number = 5500005555555559
credit_card_expiration_date_year	Year of client's payment card expiration date. length: 4 regular expression: <code>^(20)((19) ([2-9][0-9]))\$</code> Example: credit_card_expiration_date_year = 2019
credit_card_expiration_date_month	Month of client's payment card expiration date. length: 2 regular expression: <code>^(0[1-9]) (1[0-2])\$</code> Example: credit_card_expiration_date_month = 02
credit_card_security_code	Client's payment card security code (CVV2/CVC2). type: <i>number</i> max length: 4 regular expression: <code>^\d{3,4}\$</code> Example: credit_card_security_code = 559

Deleting registered card might be realized with methods below:

- 1) Client might use link given in payment confirmation emails.
- 2) Deregistration request might be sent to Dotpay from seller's system via API.

Request should be sent using *DELETE* method to the `https://ssl.dotpay.pl/t2/payment_api/v1/cards/{credit_card_id}/` address where `{credit_card_id}` is card ID which should be removed.

Below is exemplary request (in PHP) and response of card deregistration. Authorization data (user, password) is adequate to seller GUI login credentials.

Request:

```

1 <?php
2
3 $ch = curl_init();
4   curl_setopt($ch, CURLOPT_URL, "https://ssl.dotpay.pl/t2/payment_api/v1/
   ↪ cards/59f92e2bf8bedc36bec221...718c58eb13d6333ce71369184eb7ab02ae/");
5   curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
6   curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
7   curl_setopt($ch, CURLOPT_CAINFO, "ca-bundle.crt"); //http://curl.haxx.se/
   ↪ docs/caextract.html
8   curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
9   curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

```

(continues on next page)

(continued from previous page)

```
10 curl_setopt($ch, CURLOPT_TIMEOUT, 100);
11 curl_setopt($ch, CURLOPT_USERPWD, 'user:password');
12 curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
13
14 $response = curl_exec($ch); // API response
15 $curl_info = curl_getinfo($ch); //curl info
16 curl_close($ch);
17
18 echo '<pre>';
19 echo 'HTTP status code: '.$curl_info[http_code];
20 echo PHP_EOL.'-----'.PHP_EOL.PHP_EOL;
21 print_r(json_decode($response));
22 echo '</pre>';
```

Sample request to unregister the card:

**DELETE** /cards/(string: credit\_card\_id)/

Response:

```
HTTP/1.1 204 No Content
```

HTTP status code meaning:

CODE	MEANING / DESCRIPTION
204 No Content <sup>17</sup>	Deleted
404 Not Found <sup>18</sup>	Credit Card not found
400 Bad Request <sup>19</sup>	Request processing error

### 4.3 Split payment (Multimerchant)

---

**Note:** Functionality is only available for shops ( id ) which were adequately configured by Dotpay, what is dependent on signed agreement.

---

Functionality allows to split one payment into multiple amounts to be accounted on given shops ( id ). Request has to contain additional parameters (required id(n) , amount(n) , and optional currency(n) , description(n) , control(n) , where (n) is a greater than 0 integer defining consecutive ID, amounts etc.) alongside standard parameters described in *ADVANCED VERSION* section.

**Caution:** It is not possible to make a Multimerchant payment with different currencies at once, value of each currency(n) parameter has to be the same as currency .

When using this functionality parameter id defines shop (called multimerchant parent) where operation payment\_multimerchant\_parent will be created, only as an information about full payment amount (sent in amount parameter). Meanwhile on shops id(n) (called multimerchant child) amounts amount1 - amount(n) will be accounted as payment\_multimerchant\_child operation.

<sup>17</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.5>

<sup>18</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

<sup>19</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1>

Information sent is URLC notifications is adequate to given payment\_multimerchant\_parent or payment\_multimerchant\_child created on each shop ( id ).

**Caution:** In order for URLC notifications to be sent on multimerchant child accounts, an address has to be entered in each id(n) configuration (*Settings -> Notifications -> Urlc configuration -> Edit*) as there is no option to define parameter like urlc(n) .

**Caution:** It is only possible to refund payment\_multimerchant\_child operation accounted on given id(n) . It is **NOT** possible to refund whole payment using payment\_multimerchant\_parent .

Below is an exemplary split payment form.

```
1 <div>
2
3   <form action="https://ssl.dotpay.pl/t2/" id="dotpay_redirection_form"
4   ↪method="POST" enctype="application/x-www-form-urlencoded">
5
6     <input type="hidden" name="id" value="123456">
7     <input type="hidden" name="amount" value="320.00">
8     <input type="hidden" name="currency" value="PLN">
9     <input type="hidden" name="description" value="Płatność za zamówienie_
10     ↪01/2017 parent">
11     <input type="hidden" name="control" value="control_parent">
12
13     <input type="hidden" name="id1" value="456123">
14     <input type="hidden" name="amount1" value="120.00">
15     <input type="hidden" name="currency1" value="PLN">
16     <input type="hidden" name="description1" value="Płatność za zamówienie_
17     ↪01/2017 child1">
18     <input type="hidden" name="control1" value="control_child1">
19
20     <input type="hidden" name="id2" value="561423">
21     <input type="hidden" name="amount2" value="90.00">
22     <input type="hidden" name="currency2" value="PLN">
23     <input type="hidden" name="description2" value="Płatność za zamówienie_
24     ↪01/2017 child2">
25     <input type="hidden" name="control2" value="control_child2">
26
27     <input type="hidden" name="id3" value="642513">
28     <input type="hidden" name="amount3" value="110.00">
29     <input type="hidden" name="currency3" value="PLN">
30     <input type="hidden" name="description3" value="Płatność za zamówienie_
31     ↪01/2017 child3">
32     <input type="hidden" name="control3" value="control_child3">
33   </form>
34
35   <p>
36     <button type="submit" form="dotpay_redirection_form" value="Submit">
37       Potwierdź zamówienie i zapłać
38     </button>
39   </p>
```

(continues on next page)



&lt;/div&gt;

**Important:** Sum of amounts sent in parameters amount1, amount2, ..., amount(n) has to be equal to the total amount of payment, defined in parameter amount .

## 4.4 Masscollect

**Note:** Functionality is only available for shops ( id ) which were adequately configured by Dotpay, what is dependent on signed agreement.

Functionality allows to send recipient bank account number in transaction request, which is going to be used in payout, that is default account number defined in account ( id ) configuration will be overwritten.

Depending on account configuration, payouts created by automatic payouts mechanism will either be aggregated by bank account numbers or in 1:1 for where every payment creates separate payout. Payouts of any amount are always aggregated.

**Caution:** Using described functionality requires request sent to Dotpay to be signed as well as signature verification to be required on given shop ( id ). Description of signature creation can be found in *Redirection parameters integrity protection (CHK)* section.

In order to make Masscollect payments, additional parameters described in the table below should be sent in the order. Only the recipient\_account\_number parameter is required, however, it is also recommended to provide the recipient's name using the parameter recipient\_company or recipient\_first\_name and recipient\_last\_name .

### 4.4.1 Table 12. (Additional parameters used in the Masscollect service)

PARAMETER	MEANING / DESCRIPTION
recipient_account_number	Recipient account number in BBAN <sup>20</sup> format. type: <i>string</i> max length: 26 Example: recipient_account_number = 32249000896640389235035459
recipient_company	Recipient company name type: <i>string</i> max length: 50 Example: recipient_company = My Company Inc.
recipient_first_name	Recipient first name type: <i>string</i> max length: 30 Example: recipient_first_name = John

Continued on next page

Tablica 13 - continued from previous page

PARAMETER	MEANING / DESCRIPTION
recipient_last_name	Recipient last name type: <i>string</i> max length: 30 Example: recipient_last_name = Smith
recipient_address_street	Street - recipient address type: <i>string</i> max length: 40 Example: recipient_address_street = Wielicka
recipient_address_building	Building number - recipient address type: <i>string</i> max length: 10 Example: recipient_address_building = 72
recipient_address_apartment	Flat number - recipient address type: <i>string</i> max length: 10 Example: recipient_address_apartment = 1
recipient_address_postcode	Postcode - recipient address type: <i>string</i> max length: 6 Example: recipient_address_postcode = 30-552
recipient_address_city	City - recipient address type: <i>string</i> max length: 50 Example: recipient_address_city = Cracow

Below is an exemplary Masscollect payment form.

```

1 <div>
2
3   <form action="https://ssl.dotpay.pl/t2/" id="dotpay_redirection_form"
4   ↪method="POST" enctype="application/x-www-form-urlencoded">
5
6     <input type="hidden" name="id" value="123456">
7     <input type="hidden" name="amount" value="123.45">
8     <input type="hidden" name="description" value="Płatność za zamówienie_
9     ↪07/2017">
10    <input type="hidden" name="recipient_account_number" value=
11    ↪"32249000896640389235035459">
12    <input type="hidden" name="recipient_company" value="Moja Firma S.A.">
13    <input type="hidden" name="recipient_first_name" value="Jan">
14    <input type="hidden" name="recipient_last_name" value="Kowalski">
15    <input type="hidden" name="recipient_address_street" value="Wielicka">
16    <input type="hidden" name="recipient_address_building" value="72">
17    <input type="hidden" name="recipient_address_apartment" value="1">
18    <input type="hidden" name="recipient_address_postcode" value="30-552">
19    <input type="hidden" name="recipient_address_city" value="Kraków">
20    <input type="hidden" name="chk" value="3135b6debcd8fe4e488ec2easux506c31867f78bed0e
21    ↪" />

```

(continues on next page)

<sup>20</sup> [https://en.wikipedia.org/wiki/International\\_Bank\\_Account\\_Number](https://en.wikipedia.org/wiki/International_Bank_Account_Number)

(continued from previous page)

```
18 </form>
19
20 <p>
21   <button type="submit" form="dotpay_redirection_form" value="Submit">
22     Potwierdź zamówienie i zapłać
23   </button>
24 </p>
25
26 </div>
```

## 4.5 Shipping and payer data

In order for channel provider to evaluate customer's credit score, new additional parameters have been added.

Below is the list of those parameters with their description and requirements.

Sending more data than just required minimum might have big impact on credit score. If it is possible, it's recommended to send complete data.

In order to process a given payment method, a new customer parameter has been added for integration with Dotpay. This parameter should contain specific information according to the following [Table 13](#). Some payment channels may require less data, such as [PayPo](#) or [Raty Alior](#).

### 4.5.1 Table 13. (Data handled by customer parameter)

FIELD NAME	TYPE	DESCRIPTION
payer.first_name	string	Payer name
payer.last_name	string	Payer lastname
payer.email	string	Payer email address
payer.phone	string	Payer phone number
payer.address	-	Payer address
payer.address.city	string	Payer address: city
payer.address.street	string	Payer address: street
payer.address.building_number	string	Payer address: building number
payer.address.flat_number	string	Payer address: flat number
payer.address.postcode	string	Payer address: post code
payer.address.country	string	Payer address: (ISO 3166-1 alpha2 <sup>21</sup> ) or (ISO 3166-1 alpha3 <sup>22</sup> ) country code
is_logged_in	boolean	Whether payer has register an account before placing an order
registered_since	string	Payer's registration date on the seller's website, format YYYY-MM-DD or YYYY-MM-DD hh:mm:ss Optional, if it's sent, order_count is also required
registered_since_indicator	string ( <i>indicator</i> )	Payer's registration date on the seller's website, indicator for the registered_since field Optional, if it's sent, order_count is also required
account_update	string	Date of last change of paying account on the seller's website, format YYYY-MM-DD

Continued on next page

Tablica 14 - continued from previous page

FIELD NAME	TYPE	DESCRIPTION
account_update_indicator	string ( <i>indicator</i> )	Date of last change of paying account on the seller's website, indicator for the field account_update
password_change	string	Date of last password change for the paying account on the seller's website, format YYYY-MM-DD
password_change_indicator	string ( <i>indicator</i> )	Date of last change of password for the paying account on the seller's website, indicator for the field password_change
shipping_address_since	string	Date from when the payer's delivery address is used, format YYYY-MM-DD
shipping_address_since_indicator	string ( <i>indicator</i> )	Date from which the payer's delivery address is used, the indicator for the field shipping_address_since
order_count	int	How many orders the payer has made since registration Optional, if it's sent, registered_since is also required
order_count_day	int	The number of orders placed by the paying seller on the same day
order_count_year	int	Number of orders placed by the paying seller in the same year
fraud_activity	boolean	Has the store ever seen suspicious activity on this buyer's account
order_history	-	Current payer order history
order_history.date	string	n-order place date in YYYY-MM-DD format
order_history.amount	decimal (10,2)	Gross price of n-product
order	-	Order
order.total_amount	string	The value of the entire order
order.id	string	Order ID in the seller's system. Corresponds to the ID number of the entire order in the store database
order.items	-	The contents of the shopping cart. We recommend that you do not exceed the number 100 articles in the list
order.items.id	string	ID of given shop product
order.items.name	string	The name of a single product in the seller's system. Corresponds to the name of the product in the store database. min length: 1 max length: 150 regular expression: <code>[\\w\\s\\-\\. ,'?@\\ \\/ąćęłńóśźżĄĆĘŁŃÓŚŹŻ]+\$</code>
order.items.quantity	int	Number of items in the customer's order (n * Goods = total number of items) min length: 1 max length: 1000
order.items.unit_type	string	Item measurement unit, eg. quantity, kg, liters, meters and so on.
order.items.gross_price	decimal(10,2)	Gross price of n-product. Corresponds to the price of the product in the store database.
order.items.type	string	Order type (eg. product, payment, delivery, discount)
order.items.is_virtual	boolean	Product is virtual (doesn't require a delivery)
order.items.category	string	The category of a single product in the seller's system. the category should match one of the elements listed in the dictionary. List of available categories: <i>Dictionary of available sales categories</i>

Continued on next page

Tablica 14 - continued from previous page

FIELD NAME	TYPE	DESCRIPTION
order.delivery_type	string	Delivery method Available values: <ul style="list-style-type: none"> <li>• COURIER - courier</li> <li>• POCZTA_POLSKA - Poczta Polska</li> <li>• PICKUP_POINT - pickup point like UPS Access point, DHL Parcel Shop</li> <li>• PACZKOMAT - parcel locker</li> <li>• PACZKA_W_RUCHU - paczka w ruchu</li> <li>• PICKUP_SHOP -pickup in shop (click&amp;collect)</li> </ul>
order.delivery_address	-	Delivery address <b>If the package is delivered to a point / parcel locker / etc, such address and name should be given, not the details of the actual recipient.</b>
order.delivery_address.city	string	Delivery address: city
order.delivery_address.street	string	Delivery address: street
order.delivery_address.building_number	string	Delivery address: building number
order.delivery_address.flat_number	string	Delivery address: flat number
order.delivery_address.postcode	string	Delivery address: post code
order.delivery_address.country	string	Delivery address: (ISO 3166-1 alpha2 <sup>23</sup> ) or (ISO 3166-1 alpha3 <sup>24</sup> ) country code
order.delivery_address.name	string	Name of recipient / collection point. Examples: order.delivery_address.name = D0B019A order.delivery_address.name = PPP:6252652
order.delivery_address.phone	string	Recipient's phone number
order.delivery_address.is_verified	bool	Delivery address: Whether the delivery address is verified

**Note:** If the store does not want to provide the correct date, it is possible to use an indicator field of replacement type for selected parameters.

#### 4.5.2 Values used for indicator field replacement for selected fields:

VALUE	DESCRIPTION
01	The payer's account does not exist on the seller's website
02	Date of the transaction just ordered
03	Date not older than 30 days ago
04	Date in the range 30 - 60 days ago

Continued on next page

<sup>21</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-2](https://en.wikipedia.org/wiki/ISO_3166-1_alfa-2)

<sup>22</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-3](https://en.wikipedia.org/wiki/ISO_3166-1_alfa-3)

<sup>23</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-2](https://en.wikipedia.org/wiki/ISO_3166-1_alfa-2)

<sup>24</sup> [https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-3](https://en.wikipedia.org/wiki/ISO_3166-1_alfa-3)

Tablica 15 - continued from previous page

VALUE	DESCRIPTION
05	Date older than 60 days ago

**Exemplary use of parameters described above:**

Listing 1: Example in json format

```

1 {
2   "payer": {
3     "first_name": "Jan",
4     "last_name": "Kowal",
5     "email": "jankowal@example.com",
6     "phone": "123456789",
7     "address": {
8       "city": "Warszawa",
9       "street": "Krucza",
10      "building_number": "1a",
11      "flat_number": "4",
12      "postcode": "00-950",
13      "country": "POL"
14    }
15  },
16  "is_logged_in": true,
17  "registered_since": "2017-02-11",
18  "order_count": 2,
19  "order_history": [
20    {
21      "date": "2017-02-11",
22      "amount": "456.21",
23      "delivery_type": "COURIER"
24    },
25    {
26      "date": "2018-05-21",
27      "amount": "879.67",
28      "delivery_type": "POCZTA_POLSKA"
29    }
30  ],
31  "order": {
32    "id": "MHH67HF8DS",
33    "items": [
34      {
35        "id": "3245623",
36        "name": "Super Phone 1",
37        "quantity": 1,
38        "unit_type": "szt.",
39        "gross_price": "856.52",
40        "type": "towar",
41        "is_virtual": false
42      },
43      {
44        "id": "3245625",
45        "name": "Dostawa",
46        "quantity": 1,
47        "unit_type": "szt.",
48        "gross_price": "15.00",
49        "type": "dostawa",

```

(continues on next page)

```

50         "is_virtual": false
51     },
52 ],
53 "delivery_type": "POCZTA_POLSKA",
54 "delivery_address": {
55     "city": "Kraków",
56     "street": "Wielicka",
57     "building_number": "28b",
58     "flat_number": "5",
59     "postcode": "30-552",
60     "country": "POL"
61 }
62 }
63 }

```

#### 4.5.3 Dictionary of article categories for the parameter `order.items.category`

VALUE	DESCRIPTION
ART_ANTIQUES	Works of art, antiques
BOOKS_EDUCATIONAL	Books, educational materials
BUILDING_MATERIALS	Building Materials
CARS_MOTORBIKES_SCOOTERS	Cars, motorbikes, mopeds, scooters
CLOTHING_SHOES	Clothing, shoes
COMP_COMPONENTS	Computer components and accessories including monitors, printers and scanners
COMPUTERS	Computers
CONSOLES_GAMES	Consoles and games
DEVOTIONAL	Devotional articles
DOORS_WINDOWS	Windows and doors
ECO_SYSTEMS	Ecological systems
FOR_CHILDREN	Baby carriages, car seats and others
FUEL_COAL	Fuel and coal
FURNITURE	Furniture
GARDENING	Seeds, fertilizers, plants
GIFT_VOUCHERS	Gift vouchers, packages (other than medical)
HEATING	Central heating boilers, JUNKERS
HOME_FURNISHINGS	Home furnishings
HOME_TEXTILES	Home textiles, bedding and blankets
HOUSEHOLD_GOODS_BIG	Household appliances large
HOUSEHOLD_GOODS_SMALL	Household appliances small
INSURANCE	Insurance services
JEWELRY_WATCHES	Jewelry, watches
LAPTOPS_TABLETS	Laptops and tablets
MUSICAL_INSTRUMENTS	Musical instruments
OTHER_MOTORIZATION	Other motorization
PHONES_GPS	Telephones and GPS
PHOTO_CAMERA	Photo and Camera
RTV	RTV
SERVICES	Services
SOFTWARE	Software and other computer components
SPORTS_REHABILITATION	Sporting / rehabilitation / tourist articles
TOOLS_DEVICES	Tools and devices

Continued on next page

Tablica 16 – continued from previous page

VALUE	DESCRIPTION
VEHICLE_ACCESSORIES_EQUIPMENT	Vehicle accessories and equipment

#### 4.5.4 Data required in customer parameter for PayPo channel

To make a payment with PayPo channel, specific data is required, which has been described in *Table 13* :

Listing 2: Minimum data required for PayPo in json:

```

1  {
2    "payer": {
3      "first_name": "Jan",
4      "last_name": "Kowal",
5      "email": "jankowal@example.com"
6    },
7    "order": {
8      "delivery_address": {
9        "city": "Kraków",
10       "street": "Wielicka",
11       "building_number": "28B",
12       "postcode": "30-552"
13     }
14   }
15 }

```

Listing 3: Complete data for PayPo in json:

```

1  {
2    "payer": {
3      "first_name": "Jan",
4      "last_name": "Kowal",
5      "email": "jankowal@example.com",
6      "phone": "+48126882600"
7    },
8    "registered_since": "2017-02-11",
9    "order_count": 2,
10   "order": {
11     "id": "MHH67HF8DS",
12     "delivery_type": "POCZTA_POLSKA",
13     "delivery_address": {
14       "city": "Kraków",
15       "street": "Wielicka",
16       "building_number": "28b",
17       "flat_number": "5",
18       "postcode": "30-552",
19       "country": "POL"
20     }
21   }
22 }

```

*Optional parameters for PayPo have been colored, their absence might negatively impact payer's credit score.*

In order for this information to be passed to Dotpay in customer parameter, it has to be formatted correctly. First it needs to be encoded in JSON<sup>25</sup> format and then encoded using

<sup>25</sup> <https://en.wikipedia.org/wiki/JSON>



Base64<sup>26</sup>.

Listing 4: Example for generating customer parameter value (minimum) for **PayPo** channel in PHP:

```
1  <?php
2
3  $customer = array (
4
5      "registered_since" => "2017-12-31",
6      "order_count" => 12,
7
8
9      "payer" => array(
10         "first_name" => "Jan",
11         "last_name" => "Kowal",
12         "email" => "jan@example.com"
13     ),
14     "order" => array(
15         "delivery_type" => "COURIER",
16         "delivery_address" => array(
17
18             "city" => "Krakow",
19             "street" => "Wielicka",
20             "building_number" => "11",
21             "flat_number" => "7",
22             "postcode" => "30-553",
23             "country" => "POL"
24         )
25     )
26 );
27
28
29
30
31 $customer_base64 = base64_encode(json_encode($customer));
32
33 ?>
```

#### 4.5.5 Data required in customer parameter for Raty Alior channel

To make a payment with Raty Alior channel, specific data is required, which has been described in *Table 13* :

Listing 5: Minimum data required for Raty Alior in json:

```
1  {
2      "payer":{
3          "first_name":"Jan",
4          "last_name":"Kowal",
5          "email":"jankowal@example.com"
6      },
7      "order":{
8          "items":[
9              {
```

(continues on next page)

<sup>26</sup> <https://en.wikipedia.org/wiki/Base64>

(continued from previous page)

```
10         "name": "Super Phone 1",
11         "quantity": 1,
12         "gross_price": "1200.00",
13         "category": "PHONES_GPS"
14     },
15     {
16         "name": "Pendrive 64GB",
17         "quantity": 4,
18         "gross_price": "50.00",
19         "category": "COMP_COMPONENTS"
20     }
21 ]
22 }
23 }
```

The value of the `order.items.category` parameter should be filled in according to the *category dictionary*.

The number of products in the `order.items` parameter is limited and should not be more than 500 items.

In order for this information to be passed to Dotpay in customer parameter, it has to be formatted correctly. First it needs to be encoded in [JSON<sup>27</sup>](#) format and then encoded using [Base64<sup>28</sup>](#).

Listing 6: Example for generating customer parameter value (minimum) for **Raty Alior** channel in PHP:

```
1  <?php
2
3  $customer = array(
4      "payer" => array(
5          "first_name" => "Jan",
6          "last_name" => "Kowal",
7          "email" => "jankowal@example.com",
8      ),
9
10     "order" => array(
11         "items" => [
12             array(
13                 "name" => "Super_
14                 ↪Phone 1",
15                 "quantity" => 1,
16                 "gross_price" =>
17                 ↪"1200.00",
18                 "category" => "PHONES_
19                 ↪GPS"
20             ),
21             array(
22                 "name" => "Pendrive_
23                 ↪64GB",
24                 "quantity" => 4,
25                 "gross_price" => "50.
26                 ↪00",
```

(continues on next page)

<sup>27</sup> <https://en.wikipedia.org/wiki/JSON>

<sup>28</sup> <https://en.wikipedia.org/wiki/Base64>

(continued from previous page)

```
23     ↪"COMPONENTS"                                "category" => "COMP_
24                                                     )
25
26                                                     ],
27
28                                                     )
29
30                                                     );
31
32
33     $customer_base64 = base64_encode(json_encode($customer));
34
35     ?>
```

#### 4.5.6 Data required in customer parameter for PayPal channel (only with PayPal SPP)

The technical requirements necessary to obtain the “Extended Seller Protection Policy (SPP)” from PayPal to buy online - shop or reception point.

If you use the seller protection program on your PayPal account and you have an appropriate agreement in this regard with PayPal, in integration with Dotpay it is necessary to send additional information.

To make a payment with PayPal in SPP channel, specific data is required, which has been described in *Table 13* :

Listing 7: An example of using the minimum required data for a PayPal channel in SPP (json format):

```
1     {
2         "payer":{
3             "first_name":"Jan",
4             "last_name":"Kowal",
5             "email":"jankowal@example.com"
6         },
7         "order":{
8             "delivery_address":{
9                 "name":"PPP:6252652",
10                "city":"Kraków",
11                "street":"Wielicka",
12                "building_number":"28B",
13                "postcode":"30-552",
14                "phone":"+48126880000",
15                "country":"PL"
16            }
17        }
18    }
```

In order for this information to be passed to Dotpay in customer parameter, it has to be formatted correctly. First it needs to be encoded in JSON<sup>29</sup> format and then encoded using Base64<sup>30</sup>.

<sup>29</sup> <https://en.wikipedia.org/wiki/JSON>

<sup>30</sup> <https://en.wikipedia.org/wiki/Base64>

Listing 8: Example for generating customer parameter value (minimum) for **PayPal** (SPP) channel in PHP:

```
1 <?php
2
3 $customer = array(
4     "payer" => array(
5         "first_name" => "Jan",
6         "last_name" => "Kowal",
7         "email" => "jan@example.com"
8     ) ,
9     "order" => array(
10        "delivery_address" => array(
11
12            "name" => "PPP:6252652",
13            "city" => "Krakow",
14            "street" => "Wielicka",
15            "building_number" => "28B",
16            "postcode" => "30-552",
17            "phone" => "+48126880000",
18            "country" => "PL"
19        )
20    )
21 );
22
23 $customer_base64 = base64_encode(json_encode($customer));
24
25 ?>
26
```

---

**Note:** In order for the sent data to be correctly interpreted and sent from Dotpay to PayPal, such fact must be reported to Dotpay ([tech@dotpay.pl](mailto:tech@dotpay.pl)) in advance for the correct configuration of the account.

---

## 4.6 Faulty redirections from the merchant's system

On Dotpay's side a shop ( id ) may be configured 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.

---

**Note:** In order to activate functionality enable *Handling error codes in URL* option available in seller administration panel in *Settings -> Shop configuration -> Edit tab*.

---

For example, while redirecting from a merchant's system to unknown payment channel number in Dotpay, for passed in the redirection parameter url : url = <https://example.com/>

redirection will be to: [https://example.com/?error\\_code=UNKNOWN\\_CHANNEL](https://example.com/?error_code=UNKNOWN_CHANNEL)

error\_code values are listed below:

PAYMENT\_EXPIRED - payment link expired or datetime defined in expiration\_date parameter has been exceeded

UNKNOWN\_CHANNEL - invalid value of parameter channel

DISABLED\_CHANNEL - payment channel is unavailable

UNKNOWN\_CURRENCY - currency code is invalid

BLOCKED\_ACCOUNT - account ( id ) is blocked

INACTIVE\_SELLER - account ( id ) is inactive

AMOUNT\_TOO\_LOW - amount is lower than minimum defined for shop

AMOUNT\_TOO\_HIGH - amount is higher than maximum defined for shop

BAD\_DATA\_FORMAT - invalid request data, e.g. invalid expiration\_date format

URLC\_INVALID - account ( id ) settings require the URLC to contain SSL (the address starts with: 'https://')

REQUIRED\_PARAMETERS\_NOT\_PRESENT - one of the required parameters is missing

MULTIMERCHANT\_INVALID\_ACCOUNT\_CONFIGURATION - one of the accounts is not properly configured for Multimerchant payments

MULTIMERCHANT\_INSUFFICIENT\_AMOUNT - sum of amount(n) parameters is not equal to amount value

MULTIMERCHANT\_WRONG\_CURRENCY - values of parameters currency and currency(n) are not equal

CREDIT\_CARD\_NOT\_ACCEPTED - credit card data has been sent, but account ( id ) configuration doesn't allow processing it

CREDIT\_CARD\_OPERATION\_TYPE\_NOT\_ACCEPTED\_CODE - parameter sent credit\_card\_operation\_type , but account settings ( id ) does not allow its processing

CREDIT\_CARD\_SECURITY\_CODE\_REQUIRED\_NOT\_ACCEPTED\_CODE - parameter sent credit\_card\_security\_code\_required , but account settings ( id ) does not allow its processing

CREDIT\_CARD\_3DS\_NOT\_ACCEPTED\_CODE - parameter sent credit\_card\_threeds , but account settings ( id ) does not allow its processing

CREDIT\_CARD\_AVS\_NOT\_ACCEPTED\_CODE - parameter sent credit\_card\_avs , but account settings ( id ) does not allow its processing

UNKNOWN\_ERROR - returned value in other case than the above mentioned

## 5 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 Dotpay payment system.

---

### Note:

In order to receive Dotpay test environment access fill in registration form available at

[https://ssl.dotpay.pl/test\\_seller/test/registration/](https://ssl.dotpay.pl/test_seller/test/registration/)

---

The address of a test environment 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/](https://ssl.dotpay.pl/test_payment/) - address of the payment form

[https://ssl.dotpay.pl/test\\_seller/](https://ssl.dotpay.pl/test_seller/) - address of the administration panel

[https://ssl.dotpay.pl/test\\_payment/payment\\_api/channels/](https://ssl.dotpay.pl/test_payment/payment_api/channels/) - payment API address (channel list)

Simulating card payment (channel 248) requires valid card data.

Table below contains few exemplary cards which might be used for that purpose. Expiration date is anything from current date to December 2030.

### 5.1 Table 14. (List of sample card numbers that can be used in the process of testing card payments)

TYPE	NUMBER	CVV2 / CVC2	3DS
Visa*	4916 9715 6289 1025	025	No
Visa*	4929 5320 2788 7670	670	Yes
MasterCard*	5498 5400 7907 4343	343	No
MasterCard*	5344 6642 8071 1026	026	Yes
Visa	4111 1111 1111 1111	111	No
Visa	4444 4444 4444 4448	448	Yes
MasterCard	5500 0055 5555 5559	559	No

\* for One-click or recurring payment testing

## 6 ADDITIONAL INFORMATION

### 6.1 Verification of the IP address

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.

**Caution:** Merchant's system should always verify IP address from which a confirmation has been sent for security reasons.

Because of PCI DSS requirement the only SSL version used by Dotpay server is *TLsv1.2* and *TLsv1.3*, with other protocol versions encrypted connection with host [ssl.dotpay.pl](https://ssl.dotpay.pl) will not be possible.

### 6.2 Security of payment integration

In order to provide the highest security standards for your transactions we recommend using all available security mechanisms.

---

**Note: Transaction security should be handled in two places:**

- *Redirecting customer from shop to Dotpay*
  - *Receiving payment confirmation from Dotpay*
- 

### 6.2.1 Redirecting customer from shop to Dotpay

Redirecting customer to Dotpay involves sending parameters defining payment details. To prevent unauthorized data manipulation in this step we recommend using data integrity protection. Parameters can be signed, this signature is passed as an additional `chk` parameter along other standard parameters.

A description of this functionality can be found in the chapter *Redirection parameters integrity protection (CHK)*.

If your service is ready to send `chk` parameter for all of the transactions please contact Dotpay to enforce its verification.

**Caution:** Keep in mind minimal amount of payment parameters sent to Dotpay is: `id` , `amount` , `currency` , `description` .

Please verify whether your integration send those parameters correctly, according to format defined in this payment implementation manual.

### 6.2.2 Receiving confirmation of the posted payment in Dotpay

In order to send operation (transaction) confirmation to shop *URLC* (HTTP request, call-back), mechanism is used, which using *POST* method passes all parameters.

If you are using this automatic notifications *URLC*, please make sure for every order received data is verified, especially when it comes to operation type ( `operation_type` ), it's status ( `operation_status` ) and: - `operation_original_amount` - The amount taken from amount parameter which has been sent by merchant's system.- `operation_original_currency` - The currency taken from currency parameter which has been sent by merchant's system. - `signature` - checksum created by SHA-256 function from concatenated notification parameters.

We also recommend to verify *URLC* notification origin, so in this case if **IP** belongs to Dotpay and is made using *POST* method. The IP address from which Dotpay sends notifications is listed in the chapter *Verification of the IP address*.

**Caution:** After receiving the confirmation of payment in the Dotpay service, the store should absolutely compare the value of the amount and currency paid with the expected amount registered when placing the order by the buyer in the store's database.

**The order should be processed by the store only when it checks whether the value due for the order stored in the store's database is equal to the value posted in Dotpay assigned to the order!**







## 7 ATTACHMENT I - PAYMENT CHANNELS

In the following table all channels which are available in Dotpay system are presented. Availability of payment method in a given account ( id ) depends on signed contract.

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.

### 7.1 Table 15. (Payment channels available in Dotpay system)

#### 7.1.1 PAYMENT CARDS

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
71	MasterPass	-		24/7	Available only for business accounts.
246	Payment cards	-		24/7	Available only for business accounts.
248	Payment cards	-		24/7	Available only for business accounts. Available for currency: PLN, EUR, USD, GBP
249	Visa SRC	-		24/7	Available only for business accounts.
260	Google Pay	Google LLC		24/7	Available only for business accounts. Required separate agreement.
262	Apple Pay	Apple Inc.		24/7	Available only for business accounts.



#### 7.1.2 FAST TRANSFERS



NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
1	mTransfer	mBank S.A.		24/7	
2	Płacę z Inteligo	Bank PKO BP		24/7	
4	Płacę z iPKO	Bank PKO BP		24/7	
6	Przelew24	Santander Bank Polska SA (previously Bank Zachodni WBK SA)		24/7	
36	Pekao24Przelew	Bank Pekao S.A.		24/7	
38	Płać z ING	ING Bank Śląski S.A.		24/7	
44	Millennium - Płatności Internetowe	Millennium Bank S.A.		24/7	
45	Pay with Alior Bank	Alior Bank S.A.		24/7	
46	Płacę z Citi Handlowy	Citi Bank Handlowy S.A.		24/7	
50	Pay Way Toyota Bank	Toyota Bank Polska		24/7	
51	Płać z BOŚ	BOŚ Bank S.A.		24/7	
66	Bank Nowy BFG S.A.	Bankowy Fundusz Gwarancyjny (dawniej Podkarpacki Bank Spółdzielczy)		24/7	
70	Pocztowy24	Bank Pocztowy S.A.		24/7	
73	BLIK	Polski Standard Płatności Sp. z o.o.		24/7	

Continued on next page

Tablica 19 - continued from previous page






NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
74	Banki Spółdzielcze	Krajowa Izba Rozliczeniowa S.A.		24/7	
75	Płacę z Plus Bank	Krajowa Izba Rozliczeniowa S.A.		24/7	
76	Getin Bank PBL	Krajowa Izba Rozliczeniowa S.A.		24/7	
80	Noble Pay	Krajowa Izba Rozliczeniowa S.A.		24/7	
81	Idea Cloud	Krajowa Izba Rozliczeniowa S.A.		24/7	
83	EnveloBank	Bank Pocztowy S.A.		24/7	
86	TrustPay	Trust Pay		24/7	Available only for business accounts. Available for currency: CZK, EUR
87	Credit Agricole PBL	Credit Agricole Bank Polska S.A.		24/7	Available only for business accounts.
90	BNP Paribas - płacę z Pl@net	Krajowa Izba Rozliczeniowa S.A.		24/7	Available only for business accounts.
91	Nest Bank	Krajowa Izba Rozliczeniowa S.A.		24/7	Available only for business accounts.
92	Bank Spółdzielczy w Brodnicy	Krajowa Izba Rozliczeniowa S.A.		24/7	Available only for business accounts.

Continued on next page


Tablica 19 - continued from previous page

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
93	Kasa Stefczyka	Spółdzielcza Kasa Oszczędnościowo-Kredytowa im. F. Stefczyka		24/7	

### 7.1.3 TRANSFERS


NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
7	ING Corporate customers	ING Bank Śląski S.A.	 klienci korporacyjni	Mon - Sat 8:00am - 8:00pm	
10	Millennium Corporate customers	Millennium Bank S.A.	 Klienci korporacyjni	Mon - Fri 8:00am - 8:00pm	
15	iPKO	Bank PKO BP	Powszechna Kasa Oszczędności Bank Polski S.A.	0:00am - 11:00pm / 7	
16	Credit Agricole	Credit Agricole Bank Polska S.A.		4:00am - 11:00pm / 7	
32	BNP Paribas	BNP Paribas Bank Polska SA		Mon - Fri 8:00am - 9:00pm	
89	Santander	Santander Bank Polska SA (previously Bank Zachodni WBK SA)		Mon - Fri 8:00am - 9:00pm	

### 7.1.4 CASH

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
11	Bank transfer / postal	-		Mon - Fri 3 Elixir sessions	

Continued on next page

Tablica 21 - continued from previous page

NUMBERNAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
82	SEPA transfer	-	 Mon - Fri 3 Elixir sessions	Available only for business accounts. Available for currency: EUR


### 7.1.5 PURSES AND VOUCHERS

NUMBERNAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
52	SkyCash	SkyCash Poland S.A.	 24/7	
59	CinkciarzPAY	Conotoxia Sp. z o.o.	 24/7	
218	paysafecard	Paysafecard	 24/7	Available for currency: PLN WARNING: refunds are not available on this channel



### 7.1.6 INSTALLMENTS

NUMBERNAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
55	Raty z Alior Bankiem	Alior Bank S.A.	 24/7	Available only for business accounts, for orders between 300.00 PLN and amount defined in the agreement.
68	mRaty	mBank S.A.	 24/7	Available only for business accounts, for orders between 300.00 PLN and amount defined in the agreement.





### 7.1.7 OTHERS

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
212	PayPal	PayPal		24/7	Available for currency: PLN Gateway model, Dotpay doesn't send funds to the seller.

### 7.1.8 POSTPONED PAYMENTS

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
94	Kupuj teraz, zapłać później [Buy now, pay later]	Aiqlabs Sp. z o.o.		24/7	Available only for business accounts, for orders between 100.00 PLN and 2000.00 PLN or amount defined in the agreement.
92	PayPo	PayPo Sp. z o.o.		24/7	Available only for business accounts. for orders between 40.00PLN and 1000.00PLN. Required separate agreement. Requires customer parameter.

### 7.1.9 MOBILE TRANSFERS - Direct Carrier Billing type

NUMBER	NAME	PROVIDER	LOGO	POSTING PAYMENTS	REMARKS
231	Orange	Orange Polska S.A.		24/7	Available only for business accounts. Required separate agreement with Dotpay partner.
232	T-Mobile	T-Mobile Polska S.A.		24/7	Available only for business accounts. Required separate agreement with Dotpay partner.
233	PLAY	P4 Sp. z o.o.		24/7	Available only for business accounts. Required separate agreement with Dotpay partner.
234	Plus	Polkomtel Sp. z o.o.		24/7	Available only for business accounts. Required separate agreement with Dotpay partner.

## 7.2 Payment Channel List - available for download

**Note:** Listed channels and their logos are available for download at:

- Channel logos: 1.78.22.1: [format zip](#)
- Available payment methods : 1.78.22.1: [format xlsx](#)
- Available payment methods: 1.78.22.1: [format json](#)

## 8 ATTACHMENT II - OPERATION STATUSES DESCRIPTION

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

**8.1 Table 16. (Operation statuses created in the Dotpay system)**

STATUS	MEANING / DESCRIPTION
new (new)	A new operation. It means establishment of the operation.
processing (waiting for deposit)	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 (completed)	For payment, 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 (rejected)	For payment, payment_multimerchant_child types it means that Dotpay has <b>NOT</b> 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. For 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 (waiting for realization)	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.

Continued on next page

Tablica 27 - continued from previous page

STATUS	MEANING / DESCRIPTION
processing_realization (realisation)	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.

## 9 CHANGELOG

VERSION	DATE	CHANGES DESCRIPTION
1.78.22.1	2021-04-26	adding an additional optional parameter to <i>Receiving confirmation of the posted payment in Dotpay</i> updating the list of available payment channels
1.77.10.1	2021-02-22	changed channels logo: 71 'MasterPass', 246, 'Payment cards', 248 (Payment cards) removed channel 35 'Kantor Polski'
1.75.7.1	2020-11-30	logo change for the channel: 1 (mTransfer) adding new values for lang = lt (Lithuanian) and lang = lv (Latvian) removed channel 60 (Płace z T-Mobile Usługi Bankowe')
1.74.2.1	2020-11-16	adding an additional optional parameter to <i>URLC notification: operation_seller_code</i> add new channel group in <i>channel_groups</i>
1.73.13.1	2020-10-16	logo change for the channel: 50 (Pay Way Toyota Bank)
1.72.3.1	2020-09-15	removed channel 21 'VIA - Moje Rachunki' adding a new parameters <i>order.delivery_address.name</i> , <i>order.delivery_address.phone</i> , <i>order.delivery_address.is_verified</i> in <i>Chapter Shipping and payer data</i> adding data requirements sent in the parameter customer for the channel <i>PayPal - only for SPP (Seller Protection Policy)</i>
1.71.10.1	2020-08-18	removed channel 84 'Volkswagen Bank direct' adding an additional optional parameter to <i>URLC notification: channel_reference_id</i>
1.70.0.1	2020-07-27	logo change for the channel 55 (Raty z Alior Bankiem) and channel 249 (Visa SRC)
1.69.18.2	2020-06-26	changed channels logo and and provider name for 249 'Visa SRC' (before: 'Visa Checkout')
1.69.18.1	2020-06-19	logo change for the channel: 94 (Kupuj teraz, zapłać później)
1.67.18.2	2020-04-22	removed channel 65 (Płace z Idea Bank)
1.67.18.1	2020-04-22	adding a new parameter <i>order.items.category</i> in <i>Chapter Shipping and payer data</i> adding data requirements sent in the parameter customer for the channel <i>Raty Alior</i> adding additional optional parameters to <i>URLC CONFIRMATION: payer_bank_account_name</i> , <i>payer_bank_account</i> , <i>payer_transfer_title</i> , <i>blik_voucher_pin</i> , <i>blik_voucher_amount</i> , <i>blik_voucher_amount_used</i>
1.65.2.1	2020-02-11	changed channels logo and and provider name for 66 'Bank Nowy BFG S.A.' (before: 'Płace z PBS')

Continued on next page



Tablica 28 - continued from previous page

VERSION	DATE	CHANGES DESCRIPTION
1.62.2.1	2019-11-08	remove channels 48 (BNP Paribas - Płacę z Żółty), 88 (BNP Paribas dawni Klienci Raiffeisen), 56 (eurobank - płatność on-line) added channel 262 (Apple Pay) logo change for the channel 90 (BNP Paribas - płacę z Pl@net) extension of the customer parameter by additional fields in <i>Shipping and payer data</i> Chapter ( <i>account_update</i> , <i>account_update_indicator</i> , <i>fraud_activity</i> , <i>password_change</i> , <i>password_change_indicator</i> , <i>shipping_address_since_indicator</i> ) added parameter <i>ap_token</i>
1.60.16.1	2019-09-19	removed channel 33 (Volkswagen Bank)
1.59.10.2	2019-08-12	adding a new parameter <i>order.id</i> in Chapter <i>Shipping and payer data</i> , adding parameter description <i>pid</i> typographic corrections
1.59.10.1	2019-07-12	logo change for the channel 51 (Płać z BOS) and channel 95 (PayPo)
1.58.2.1	2019-06-10	changed channels logo and and provider name for 55 'Raty z Alior Bankiem' (before: 'erata - raty z dotpay')
1.58.0.2	2019-06-03	changed channels logo and and provider name for 93 'Kasa Stefczyka' (before: 'eSKOK') adding a new Chapter: <i>Security of payment integration</i> ; typographical corrections
1.58.0.1	2019-05-23	typographical corrections
1.56.14.1	2019-04-24	added new value for parameter <i>currency</i> : BGN, CHF, HRK, HUF, RUB
1.56.11.3	2019-04-10	removed channel 72 (Płacę z Orange) added parameter <i>gp_token</i>
1.56.11.2	2019-04-01	changed names and logos for channels: 32 (previously: BGŻ BNP Paribas), 48 (previously: R - Przelew), 88 (previously: Raiffeisen), 90 (previously: BGŻ BNP Paribas) added parameter <i>customer</i> and corresponding <i>Shipping and payer data</i> section
1.56.11.1	2019-03-28	added channel 95 (PayPo) added channel 260 (Google Pay) added current channel list in <i>xlsx</i> file for download changed names and logos for channels: 32 (previously: BGŻ BNP Paribas), 48 (previously: R - Przelew), 88 (previously: Raiffeisen), 90 (previously: BGŻ BNP Paribas)
1.55.8.1	2019-03-22	removed channel 31 (Zapłać w Żabce i we Freshmarket) removed channel 24 (mPay)
1.55.7.1	2019-02-15	added new value for parameter <i>operation_type</i> = <i>pay-out_commission</i> added new value for parameter <i>lang</i> = <i>uk</i> (Ukrainian) added new value for parameter <i>currency</i> = <i>NOK</i> removed alternative payment parameters names: <i>kwota</i> , <i>waluta</i> , <i>opis</i> , <i>kanal</i> , <i>blokuj</i> , <i>grupykanalow</i> , <i>typ</i> , <i>txtguzik</i> , <i>data_waznosci</i> , <i>forename</i> , <i>imie</i> , <i>nazwisko</i> , <i>surname</i> , <i>ulica</i> , <i>budynek</i> , <i>lokal</i> , <i>mieszkanie</i> , <i>addr2</i> , <i>miasto</i> , <i>kod</i> , <i>telefon</i> , <i>kraj</i> , <i>jezyk</i>
1.53.5.1	2018-12-10	typographical corrections added parameter <i>credit_card_unique_identifier</i> in <i>URLC</i> notifications

Continued on next page

Tablica 28 - continued from previous page

VERSION	DATE	CHANGES DESCRIPTION
1.52.6.1	2018-11-13	removed channel 58 („Szybkie Płatności Internetowe z Deutsche Bank PBC”) added a regular expression for the parameter amount
1.51.0.1	2018-10-03	added parameters in <i>URLC</i> notifications: <code>credit_card_expiration_year</code> and <code>credit_card_expiration_month</code>
1.50.11.1	2018-09-27	added channel 59 („CinkciarzPAY”) change of channel provider name from Volkswagen Bank Polska S.A. to Volkswagen Bank GmbH
1.50.8.2	2018-09-07	logo change for the channel 6 (Przelew24) and channel 89 (BZWBK)
1.50.8.1	2018-09-05	Newly created accounts now require <code>chk</code> parameter by default changes in descriptions regarding verification of order amount and currency
1.49.11.1	2018-06-26	added values O - Postponed Payments M - Mobile Transfers (DCB) for parameter <code>channel_groups</code> (grupykanałów) added channel 231 (Orange) added channel 232 (T-Mobile) added channel 233 (PLAY) added channel 234 (Plus) added channel 94 (Kupuj teraz, zapłać później) added information about TLSv1.2
1.45.2.1	2018-03-30	changed name of section One-click payments to One-click and recurring payments added new parameters to the section mentioned above and included them in Redirection parameters integrity protection (CHK) section as well added new values for <code>error_code</code> parameter described in “Faulty redirections from the merchant’s system” section
1.44.12.2	2018-03-08	added channel 93 (eSKOK)
1.44.12.1	2018-02-20	logo change for the channel 48 (R-Przelew) and 246, 248 (Payment cards)
1.44.10	2018-02-13	added new values for parameter <code>currency</code> (waluta) added new values for parameter <code>language</code> (jezyk) added exemplary cards to TEST ENVIRONMENT chapter
1.39.2.1	2017-09-03	added channel 83 (EnveloBank) added channel 249 (Visa Checkout) changed channels logo 246 (Payment cards via Payeezy) and 248 (Payment cards)
1.38.1.1	2017-08-21	added channel 15 (iPKO)
1.37.3.3	2017-07-10	added parameter in <i>URLC</i> notifications: <code>is_completed</code>
1.37.3.2	2017-06-28	added Masscollect subchapter
1.37.3.1	2017-06-27	added Split payment (Multimerchant) subchapter added Multimerchant and Surcharge parameters to Redirection parameters integrity protection (CHK) subchapter updated Faulty redirections from the merchant’s system subchapter added parameter <code>ignore_last_payment_channel</code>
1.36.10.1	2017-06-12	logo change for the channel 36 („Pekao24Przelew”)

Continued on next page

Tablica 28 - continued from previous page

VERSION	DATE	CHANGES DESCRIPTION
1.36.7.1	2017-06-01	added channel 90 (BGŻ BNP Paribas) added channel 91 (Nest Bank) added channel 92 (Bank Spółdzielczy w Brodnicy) added information about downloading payment channel list for given ID
1.35.4.2	2017-04-05	added value UAH for parameter currency (waluta) added channel 88 (Raiffeisen) added channel 89 (BZWBK)
1.35.4.1	2017-03-28	removed kanału 18 (Przelew z BPH)
1.34.9.3	2017-01-26	added parameter in <i>URLC</i> notifications: <code>credit_card_registration</code>
1.34.9.2	2017-01-11	added regular expressions changed channels logo: 71 („MasterPass”), 246 („Payment cards via Payeezy”), 248 (Payment cards)
1.34.9.1	2017-01-02	removed channel 77 (FerBuy) removed value O - postponed payments from parameter <code>channel_groups</code> (grupykanałów)
1.33.4.2	2016-12-06	added channel 218 (paysafecard)
1.33.4.1	2016-11-14	removed channel 27 (BGŻ)
1.32.6.2	2016-11-07	added <code>credit_card_registration</code> parameter removed channel 63 (Płacę z IKO)
1.32.6.1	2016-09-23	added channel 84 (Volkswagen Bank direct) added channel 86 (TrustPay) added channel 87 (Credit Agricole PBL)
1.30.6.3	2016-06-22	added <code>deladdr</code> parameter
1.30.6.2	2016-06-17	typographical corrections added <code>bylaw</code> , <code>personal_data</code> parameters added <code>credit_card_number</code> , <code>credit_card_expiration_date_year</code> , <code>credit_card_expiration_date_month</code> , <code>credit_card_security_code</code> parameters new way of receiving test environment access (added registration link)
1.30.6.1	2016-06-01	typographical corrections added One-click payments section added <code>expiration_date</code> parameter added value <code>HASH_NOT_EQUAL_CHK</code> for parameter <code>error_code</code> described in Faulty redirections from the merchant's system section
1.29.11.1	2016-03-21	added chapter ADDITIONAL FEATURES added subchapter Redirection parameters integrity protection (CHK) moved subchapter Faulty redirections from the merchant's system to chapter ADDITIONAL FEATURES
1.29.8.1	2016-02-26	added parameter <code>channel_groups</code> added information about <i>HTTPS verify</i> and <i>SSL certificate verify</i> in II. RECEIVING PAYMENT CONFIRMATION ( <i>URLC CONFIRMATION</i> ) chapter added channel 82 (Przelew SEPA) added channel 248 (Payment cards) removed channel 64 (PeoPay)
1.28.5.2	2016-01-07	added parameter <code>blik_code</code> added chapter ATTACHMENT II - OPERATION STATUSES DESCRIPTION
1.28.5.1	2015-12-31	removed channel 79 (Open Pay)

Continued on next page

Tablica 28 - continued from previous page

VERSION	DATE	CHANGES DESCRIPTION
1.27.0.1	2015-11-10	removed channel 25 (Plus Bank)
1.25.7.1	2015-10-30	removed channel 49 (MeritumBank)
1.25.3.1	2015-10-22	removed channel 22 (Ukash) removed channel 43 (Bank Spółdzielczy we Wschowie) removed channel 62 (DNB Nord)
1.25.1.1	2015-10-19	change of the main payment site (from <a href="https://ssl.dotpay.pl">https://ssl.dotpay.pl</a> to <a href="https://ssl.dotpay.pl/t2/">https://ssl.dotpay.pl/t2/</a> ) removed channel 3 (MultiTransfer)
1.24.9.1	2015-10-12	added channel 81 (Idea Cloud)
1.23.13.3	2015-09-24	typographical corrections
1.23.13.2	2015-08-20	removed channel 69 (V.me)
1.23.13.1	2015-08-12	added channel 79 (Open Pay) added channel 80 (Noble Pay) removed channel 15 (iPKO)
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	added channel 77 (FerBuy)
1.22.9.1	2015-06-01	added channel 74 (Banki Spółdzielcze) added channel 75 (Płacę z Plus Bank) added channel 76 (Getin Bank PBL) removed channel 17 (Płacę z iPKONET) removed channel 57 (Getin Bank) added parameters operation_withdrawal_amount, operation_commission_amount, channel_country, geoip_country in URLC notifications
1.20.9.2	2015-02-09	added channel 73 (BLIK)
1.20.9.1	2015-01-14	removed channel 245 (MasterCard Mobile) added chapter CHANGE LOG.
1.19.15.2	2014-12-12	added channel 72 (Płacę z Orange)
1.19.15.1	2014-12-08	added channel 71 (MasterPass)
1.18.5.4	2014-11-04	added channel 69 (V.me)
1.18.5.3	2014-11-19	removed channels 14 (KB24) and 61 (Bank Pocztowy)
1.18.5.2	2014-10-28	added channels 66 (Płacę z PBS ) and 70 (Pocztowy24) renamed channel 17 (from Płać z Nordea to Płacę z IPKOnet) added <a href="https://ssl.dotpay.pl/test_seller/">https://ssl.dotpay.pl/test_seller/</a> address to TEST ENVIRONMENT chapter

## HTTP Routing Table

/cards

DELETE /cards/(string:credit\_card\_id)/,  
52