



# **Dotpay**

## **Instrukcja techniczna implementacji płatności**

**wydanie: 1.94.1.0 pl**

**Dotpay Development Team**

Utworzono : 19 grudnia 2022

*Ta dokumentacja dostępna jest również online  
w formacie [html](#)*

# Contents

<b>1</b>	<b>Wersja podstawowa</b>	<b>1</b>
1.1	Przykładowe formularze płatności / dotacji	2
<b>2</b>	<b>Wersja zaawansowana</b>	<b>8</b>
2.1	Diagram 1. Przykładowy przebieg procesu płatności przedstawia poniższy schemat oraz opis:	9
2.2	Tabela 1. (Podstawowe parametry przesyłane do serwisu Dotpay)	10
2.3	Tabela 2. (Dodatkowe parametry przesyłane do serwisu Dotpay)	12
<b>3</b>	<b>ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)</b>	<b>23</b>
3.1	Tabela 3. (Parametry wysyłane przez serwis Dotpay po wykonaniu operacji (transakcji))	25
3.2	Przykład liczenia parametru signature	33
<b>4</b>	<b>DODATKOWE FUNKCJONALNOŚCI</b>	<b>34</b>
4.1	Ochrona integralności parametrów przekierowania (CHK)	34
4.1.1	Przykład 1 podstawowy	38
4.1.2	Przykład 2 rozbudowany	42
4.2	One-click i płatności cykliczne	52
4.2.1	Tabela 4. (Parametry związane z rejestracją karty płatniczej)	53
4.2.2	Tabela 5. (Parametry wykorzystywane w procesie kolejnej płatności zarejestrowaną wcześniej kartą)	53
4.2.3	Tabela 6. (Parametry wspólne dla pierwszej oraz kolejnej płatności)	54
4.2.4	Przykładowe modele integracji oraz związane z nimi podstawowe wymagania	56
	Model integracji: 1 click	56
	Model integracji: recurring	56
	Model integracji mieszany: 1 click + recurring	57
4.2.5	Tabela 11. (Parametry dla danych karty płatniczej)	59
4.3	Płatność dzielona (Multimerchant / Pasaż)	61
4.4	Masscollect	62
4.4.1	Tabela 12. (Parametry dodatków wykorzystywane w usłudze Masscollect)	63
4.5	Obsługa danych dostawy oraz płacącego	65
4.5.1	Tabela 13. (Dane obsługiwane przez parametr customer)	65
4.5.2	Wartości używane dla pola zastępczego typu indyktor dla wybranych pól:	68
4.5.3	Słownik kategorii artykułów dla parametru order.items.category	69
4.5.4	Wymagane dane przesyłane w parametrze customer dla kanału <b>PayPo</b>	70
4.5.5	Wymagane dane przesyłane w parametrze customer dla kanału <b>Raty Alior</b>	71
4.5.6	Wymagane dane przesyłane w parametrze customer dla kanału <b>PayPal</b> (tylko z programem SPP w PayPal)	73
4.6	Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy	74
<b>5</b>	<b>ŚRODOWISKO TESTOWE</b>	<b>75</b>
5.1	Tabela 14. (Lista przykładowych numerów kart, które można wykorzystać w procesie testowania płatności kartowych)	76
<b>6</b>	<b>INFORMACJE DODATKOWE</b>	<b>76</b>
6.1	Weryfikacja adresu IP	76
6.2	Bezpieczeństwo integracji płatności	77
6.2.1	Przekierowanie kupującego do Dotpay	77
6.2.2	Odbiór potwierdzenia o zaksięgowanej wpłacie w serwisie Dotpay	77
<b>7</b>	<b>ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI)</b>	<b>78</b>
7.1	Tabela 15. (Kanały płatności dostępne w serwisie Dotpay)	78

7.1.1	KARTY PŁATNICZE . . . . .	78
7.1.2	SZYBKE TRANSFERY . . . . .	79
7.1.3	PRZELEWY ONLINE . . . . .	81
7.1.4	PŁATNOŚCI GOTÓWKOWE . . . . .	82
7.1.5	PORTMONETKI I VOUCHERY . . . . .	82
7.1.6	RATY . . . . .	82
7.1.7	INNE . . . . .	83
7.1.8	PŁATNOŚCI ODROZONE . . . . .	83
7.1.9	PŁATNOŚCI MOBILNE typu Direct Carrier Billing . . . . .	84
7.2	Kanały płatności - do pobrania . . . . .	85
<b>8</b>	<b>ZAŁĄCZNIK II (OPISY STATUSÓW OPERACJI)</b>	<b>85</b>
8.1	Tabela 16. (Statusy operacji tworzone w systemie Dotpay) . . . . .	85
<b>9</b>	<b>DZIENNIK ZMIAN</b>	<b>86</b>
	<b>HTTP Routing Table</b>	<b>92</b>

## Contents

---

W systemie płatności elektronicznych Dotpay można wydzielić dwie metody przyjmowania płatności od klientów. Pierwsza (wersja podstawowa) przeznaczona jest głównie dla sprzedawców, którzy nie posiadają sklepów lub serwisów zintegrowanych z systemem Dotpay.

Druga metoda (wersja zaawansowana) przeznaczona jest dla sklepów oraz serwisów, na których rozpoczyna się proces płatności (np. klient składa zamówienie na towar).

Adresy, pod którymi dostępna jest bramka płatności Dotpay:

- środowisko produkcyjne : <https://ssl.dotpay.pl/t2/>
- środowisko testowe : [https://ssl.dotpay.pl/test\\_payment/](https://ssl.dotpay.pl/test_payment/)

**Ostrzeżenie:** Ze względu na politykę bezpieczeństwa, bramka płatności Dotpay NIE może być osadzana w ramce / iframe (X-Frame-Options:SAMEORIGIN).

## 1 Wersja podstawowa

Aby przekierować klienta do płatności, wszystkie dane można zapisać w adresie URL - linku. W poniższym przykładzie do formularza płatności przekazywane jest id sklepu, amount, currency i description:

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

Powyższy adres przekierowuje za pomocą metody *GET*.

Opis pozostałych parametrów, które można w ten sposób przekazać do formularza płatności Dotpay, znajduje się w dalszej części instrukcji, w *tabeli 1* oraz *tabeli 2*.

Po zalogowaniu w serwisie Dotpay można skorzystać z menu *Narzędzia* -> *Generator linków płatniczych*. Menu generowania linków pozwala na określenie danych takich jak: id, amount, currency, description oraz lang. Wygenerowanie linku do płatności oznacza utworzenie specjalnego tokenu, którego wywołanie pozwoli na odgórne zdefiniowanie wymienionych parametrów, a tym samym uniemożliwi modyfikację danych płatności.

Wygenerowane w ten sposób tokeny można przeglądać oraz modyfikować (np. amount) w powyżej wymienionym menu.

Przykład tokenu: rfhu4jb5ym657g3xluf4bbqfmbj6t17

Dla tak określonego tokenu stronę płatności można wywołać korzystając z następującego linku:

<https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17>

W przypadku, gdy **NIE** zostanie zdefiniowany język płatności, lub gdy wskazany uprzednio język zostanie usunięty, przesłanie dodatkowego parametru lang pozwoli określić język, w jakim wyświetlona ma zostać strona płatności.

Poniżej podany jest przykład wywołania strony płatności z dodatkowym parametrem lang = en w celu zaprezentowania formularza w języku angielskim:

<https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&lang=en>

Do linku płatniczego dodatkowo można dołączyć parametr `ignore_last_payment_channel = 1` w celu zignorowanie ostatnio wybranej przez klienta metody płatności (zapamiętanej w pamięci jego przeglądarki), dzięki czemu będzie on widział zawsze pełną listę dostępnych dla niego metod płatności.

[https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&ignore\\_last\\_payment\\_channel=1&la](https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&ignore_last_payment_channel=1&la)

Przygotowany link z tokenem można przykładowo przesłać w wiadomości e-mail do kupującego.

Utworzony link płatniczy może być wykorzystywany wielokrotnie, aż do momentu usunięcia go przez sprzedawcę w panelu Dotpay (*Narzędzia -> Generator linków płatniczych -> Usuń*).

**Ostrzeżenie:** Domyślnie dla sklepu wymagane jest przesłanie również parametru `chk`. W przypadku generowania linków płatniczych z poziomu panelu administracyjnego, parametr `chk` dodawany jest automatycznie do linku.

Natomiast, gdy link płatniczy generowany jest ręcznie z parametrów lub za pomocą API panelu administracyjnego, konieczne jest wygenerowanie poprawnej wartości `chk` oraz dołączenie jej do linku płatniczego.

Konstrukcja przykładowego linku może wyglądać tak:

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

Opis sposobu wyliczenia parametru `chk` znajduje się w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*.

**Informacja:** W przypadku, kiedy sklep nie korzysta z automatycznych notyfikacji *URLC* i potwierdzenie każdej transakcji odbywa się po manualnej weryfikacji poprawności zaksięgowanej kwoty oraz waluty dla konkretnego zamówienia, wymagalność parametru `chk` może zostać wyłączona. W tym celu prosimy o kontakt drogą mailową na adres [administracja@dotpay.pl](mailto:administracja@dotpay.pl)

## 1.1 Przykładowe formularze płatności / dotacji

**Ostrzeżenie:** Do poprawnego funkcjonowania, poniższe formularze wymagają wcześniejszej modyfikacji parametrów określonych w sekcji *KONFIGURACJA*, przykładowo wstawienia odpowiedniego id konta, na rzecz którego ma zostać wykonana płatność (Numer id można znaleźć po zalogowaniu do panelu administracyjnego w zakładce *Ustawienia*, jest to 6-cyfrowa liczba umieszczona po znaku # w kolumnie *Sklep*). W innym przypadku nastąpi przekierowanie na stronę z komunikatem błędu.

Po kliknięciu w przycisk, klient zostanie przekierowany do formularza płatności Dotpay (<https://ssl.dotpay.pl/t2/>) w celu wybrania metody płatności. Równocześnie metodą *POST* przesyłane są poniższe parametry:

- a) id sklepu sprzedawcy (np. `id = 123456`)
- b) kwota transakcji (np. `amount = 12.42`)
- c) waluta transakcji (np. `currency = PLN`)
- d) opis transakcji (np. `description = Zapłata za fakturę VAT 12345/2014`)
- e) typ przekierowania (np. `type = 0`)

f) adres powrotny po płatności (np. url = https://www.example.com)

g) tekst przycisku powrotnego (np. buttontext = Powrót)

**Ostrzeżenie:** Poniższe formularze nie zawierają funkcji wyliczającej obowiązkowy parametr chk . Funkcjonalność tą należy uzupełnić we własnym zakresie, bądź skontaktować się z działem administracji Dotpay (administracja@dotpay.pl) w celu wyłączenia weryfikacji chk . Wcześniej zapoznaj się z rozdziałem *Bezpieczeństwo integracji płatności*.

Formularz może również przesłać inne, opcjonalne parametry, których opis znajduje się w tabelach umieszczonych w dalszych częściach dokumentacji.

### 1. Formularz z predefiniowanymi oraz dowolną kwotą<sup>1</sup>

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6 </head>
7
8 <body>
9   <!-- -----
10
11   Górna część strony
12
13   ----->
14
15   <!-- copy start-->
16   <style>
17     div.dotpay_form_donation {
18       font-family: sans-serif;
19       text-align: center;
20     }
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;
```

(ciąg dalszy na następnej stronie)

<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)

```

45     background: #f8f6fb;
46     text-align: center;
47
48 }
49
50 input#dp_other_amount {
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
72 table.tbl_center {
73     margin-left: auto;
74     margin-right: auto;
75 }
76 </style>
77
78 <!-- poniżej wpisz prawidłową ścieżkę do jquery -->
79 <script type="text/javascript" src="http://code.jquery.com/jquery-3.5.1.min.js
↵"></script>
80
81 <script>
82     $(document).ready(function () {
83         $("#dp_buttonDarowizna").click(function () {
84             if ($('#dp_kwota').val().trim() === '') {
85                 $("#dp_kwota_alert").text("Proszę wybrać lub wprowadzić kwotę_
↵darowizny.").show();
86                 $('#dp_kwota_alert').css("display", "inline").fadeOut(5000);
87                 return false;
88             }
89         });
90     });
91 </script>
92
93 <div class="dotpay_form_donation">
94     <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
95         <div class="dp_temat">Wybierz kwotę darowizny</div>
96         <p>
97             <input type="button" id="dp_def_amount"
98                 onClick="$('#dp_kwota').val('10'); $('#dp_kwota').prop(
↵'readonly', true);$('#dp_kwota').attr('style','color:blue');$('#dp_other_amount_
↵txt').html('Wybrana Kwota')"
99                 value="10.00 zł" />
100             <input type="button" id="dp_def_amount"
101                 onClick="$('#dp_kwota').val('20'); $('#dp_kwota').prop(
↵'readonly', true);$('#dp_kwota').attr('style','color:blue');$('#dp_other_amount_
↵txt').html('Wybrana Kwota')"

```

(ciąg dalszy na następnej stronie)

```

102     value="20.00 zł" />
103     <input type="button" id="dp_def_amount"
104     onClick="$('#dp_kwota').val('50'); $('#dp_kwota').prop(
↳ 'readonly', true); $('#dp_kwota').attr('style','color:blue'); $('#dp_other_amount_
↳ txt').html('Wybrana Kwota')"
105     value="50.00 zł" />
106     <input type="hidden" name="type" value="0" />
107     <input type="hidden" name="currency" value="PLN" />
108
109     <table class="tbl_center">
110         <tr>
111             <td>
112                 <br><input type="button" id="dp_other_amount"
113                 onClick="$('#dp_kwota').prop('readonly', false); $(
↳ '#dp_kwota').attr('style','color:brown'); $('#dp_other_amount_txt').html('<span_
↳ style='color:brown'>Wprowadź kwotę</span>')"
114                 value="Inna kwota" />
115             </td>
116             <td>
117                 <br><span id="dp_other_amount_txt">Wybrana Kwota</span>
↳ :
118                 <input type="text" name="amount" id="dp_kwota" readonly
119                 pattern="^([1-9])(\\.\\d{1,2})?|^((?!0)(\\d){1,5}
↳ )(\\.\\d{1,2})?|^(1(\\d{5})(\\.\\d{1,2})?)|^((200000|[0-9]{1,2})?)$"
120                 placeholder="np. 100" maxlength="9" size="9"
121                 title="Kwota powinna zawierać się w przedziale 1 -_
↳ 200000 PLN. Dozwolony format to np: 100 lub 152.43"
122                 oninput="this.value = this.value.replace(/^[^0-9\\.]/
↳ g, ''); this.value = this.value.replace(/(\\.*)\\.\\/g, '$1');" />
123                 PLN
124                 <br />
125             </td>
126         </tr>
127     </table>
128
129     <!------- KONFIGURACJA ----->
↳ ----->
130
131     <!--- zamiast 000000 należy podstawic numer ID w Dotpay -->
132     <input type="hidden" name="id" value="000000" />
133     <!--- Tytuł transakcji --->
134     <input type="hidden" name="description" value="Testowa płatność" />
135     <!--- Ardes URL powrotu --->
136     <input type="hidden" name="url" value="http://www.example.com" />
137     <!--- Tekst przycisku powrotu do sklepu --->
138     <input type="hidden" name="buttontext" id="buttontext" value=
↳ "Powrót do sprzedawcy" />
139     </p>
140
141     <!------- KONIEC KONFIGURACJI ----->
↳ ----->
142
143     <p><br><button class="dp_buttomDarowizna" id="dp_buttomDarowizna">
↳ Wpłać darowiznę</button></p>
144     </form>
145     <div id="dp_kwota_alert" style="color:red;"></div>
146 </div>
147
148 <!--- copy end--->
149
150 <!------->

```





```

48 ----->
49
50 </body>
51
52 </html>

```

### 3. Formularz z dowolnym opisem i kwotą<sup>3</sup>

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="UTF-8">
6  </head>
7
8  <body>
9    <!------->
10
11     Górna część strony
12
13 ----->
14
15 <!-- copy start-->
16
17 <div style="text-align: center;">
18   <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
19     <p style="font-size: 18px">Wybierz kwotę darowizny</p>
20     <input name="description" value="Darowizna na cele statutowe" type="hidden">
21     <input name="amount" id="kwota" size="6" value="" type="text" required
22       pattern="^([1-9])(\.\d{1,2})?|^((?!0)(\d){1,5})(\.\d{1,2})?|^(1(\d{5}
↳)(\.\d{1,2})?)$|^((200000)(\.[0]{1,2})?)$"
23       placeholder="np. 10" maxlength="9" size="9"
24       title="Kwota powinna mieścić się w przedziale 1 - 200 000 PLN. Dozwolony_
↳format to np: 10 lub 10.00">
25
26     <input type="hidden" name="currency" value="PLN" /> PLN
27
28     <!------- KONFIGURACJA ----->
↳----->
29
30     <!--- zamiast 000000 należy podstawic numer ID w Dotpay -->
31     <input name="id" value="000000" type="hidden">
32
33     <!------- KONIEC KONFIGURACJI ----->
↳----->
34
35     <p><br><button class="buttonDarowizna">Wpłać darowiznę</button></p>
36   </form>
37   <br>
38 </div>
39
40 <!-- copy end-->
41
42 <!------->
43
44     Dolna część strony
45
46 ----->
47 </body>

```

(ciąg dalszy na następnej stronie)

<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)

```
</html>
```

## 2 Wersja zaawansowana

Wersja zaawansowana polega na integracji serwisu sprzedawcy z systemem płatności Dotpay. W tej wersji klient po złożeniu zamówienia i kliknięciu w przycisk potwierdzający chęć zapłaty (np. *Zapłać przez Dotpay*) zostaje przekierowany z serwisu sprzedawcy do formularza płatności Dotpay, który znajdujący się pod adresem *https://ssl.dotpay.pl/t2/*.

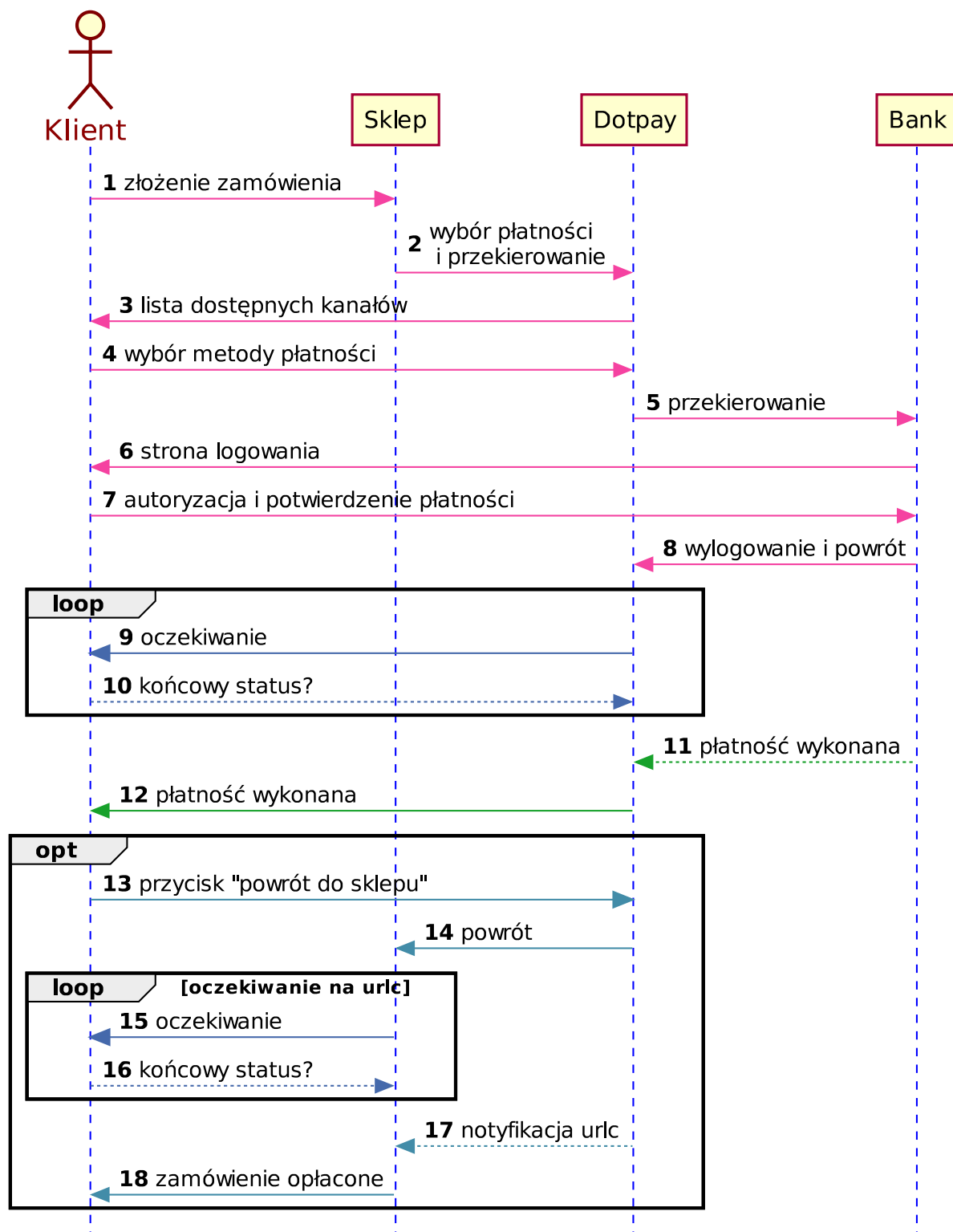
W następnym kroku klient dokonuje wpłaty wybranym kanałem płatności. Jeśli został zdefiniowany adres powrotu, oraz odpowiednia wartość parametru *type*, na stronie z potwierdzeniem płatności wyświetlony zostanie przycisk umożliwiający powrót do serwisu sprzedawcy.

Sprzedawca może również zdefiniować adres *URLC*, na który metodą *POST* będą przesyłane informacje o statusie transakcji. Adres taki można zdefiniować po zalogowaniu do serwisu Dotpay (*Ustawienia -> Powiadomienia -> Konfiguracja urlc -> Edycja*) lub przesyłać podczas inicjowania płatności jako parametr o nazwie *urlc*.

W drugim przypadku należy dla danego sklepu odblokować przyjmowanie parametru *urlc* ze źródeł zewnętrznych tj. odznaczyć opcję: *Blokuj zewnętrzne urlc (Ustawienia -> Powiadomienia -> Konfiguracja urlc -> Edycja)*. Dzięki takiej konfiguracji dla każdej płatności można przesyłać inny adres *URLC*.

Zasada działania notyfikacji *URLC* została opisana w sekcji *POWIADOMIENIA URLC*

## 2.1 Diagram 1. Przykładowy przebieg procesu płatności przedstawia poniższy schemat oraz opis:



1. Kupujący składa zamówienie w sklepie
2. po skompletowaniu koszyka klient wybiera płatność z Dotpay i jest przekierowany na formatkę płatności
3. gdzie zaprezentowana jest lista kanałów.

4. Klient dokonuje wyboru
5. i zostaje przekierowany do banku.
6. Pokazuje się strona logowania do bankowości
7. gdzie wprowadza dane autoryzacyjne i potwierdza przelew.
8. Po wylogowaniu następuje powrót na stronę Dotpay.
9. W oczekiwaniu na potwierdzenie
10. przeglądarka cyklicznie odpytuje o status płatności.
11. Bank informuje Dotpay o końcowym statusie płatności
12. i informacja ta prezentowana jest płatczemu.

---

**Informacja:** Kroki 13 - 18 są opcjonalne (w zależności od modelu integracji) i nie mają wpływu na proces płatności.

---

13. Kupujący klika na przycisk powrotny
14. i zostaje przekierowany do sklepu
15. W oczekiwaniu na potwierdzenie
16. przeglądarka cyklicznie odpytuje o status płatności.
17. Po odebraniu notyfikacji URLC
18. sklep potwierdza klientowi opłacenie zamówienia.

## **2.2 Tabela 1. (Podstawowe parametry przesyłane do serwisu Dotpay)**

PARAMETR	ZNACZENIE / OPIS
api_version	<p>Parametr domyślnie wymagany.  Parametr określający wersję API, zgodnie z którą system prześle powiadomienie URLC (opis powiadomień URLC zawarty jest w rozdz. <i>ODBIERANIE INFORMACJI PO PŁATNOŚCIACH (POWIADOMIENIA URLC)</i> ).  Przesłanie parametru nadpisuje domyślną konfigurację sklepu ( id ) ustawioną w panelu administracyjnym sprzedawcy, w zakładce <i>Ustawienia -&gt; Konfiguracja sklepu -&gt; Edycja</i>.  Dostępne wartości:  next - najnowsza wersja API opisana w niniejszej instrukcji.  dev - starsza wersja API opisana w <a href="#">instrukcji v1.78.22.1<sup>4</sup></a> .</p> <hr/> <p><b>Ważne:</b> Jeżeli konfiguracja sklepu (<i>Ustawienia -&gt; Konfiguracja sklepu -&gt; Edycja</i>) wskazuje na wersję API inną niż next, to suma kontrolna liczona dla parametru chk zostanie potraktowana jako błędnie wyliczona - niezgodnie z niniejszą dokumentacją. Tym samym żądanie zostanie odrzucone, czyli system odpowie błędem i nie będzie dalej przetwarzał takiego żądania. W logach dostępnych w panelu administracyjnym sprzedawcy Dotpay, w zakładce <i>Logi</i> dla takiego zdarzenia powinien pojawić się wpis o treści „<i>Required Chk obtained, but hash not equal.</i>”  Nie dotyczy to sytuacji gdy w konfiguracji sklepu zezwolono na nadpisanie ustawień api w konfiguracji sklepu ( id ) poprzez przesłanie w żądaniu parametru api_version = next .  <b>Zalecamy ustawienie w konfiguracji sklepu</b> (<i>Ustawienia -&gt; Konfiguracja sklepu -&gt; Edycja</i>) <b>wersję API next !</b>  W innym wypadku zalecamy przesyłanie tego parametru api_version = next w każdym żądaniu.</p> <hr/> <p>Przykład:  api_version = next</p>
id	<p>ID sklepu sprzedawcy w systemie Dotpay, na rzecz, którego dokonywana jest płatność.  Numer ID można znaleźć po zalogowaniu do panelu administracyjnego w zakładce <i>Ustawienia</i>, jest to 6-cyfrowa liczba umieszczona po znaku # w kolumnie <i>Sklep</i>.  typ: <i>integer</i>  minimalna wartość: 1  maksymalna wartość: 999999  Przykład:  id = 123456</p>
amount	<p>Kwota transakcji podana z częścią setną (zawsze dwa miejsca po separatorze). Separatorem części setnej jest znak kropki.  typ: <i>string</i>  minimalna długość: 1  maksymalna długość: 10  przykładowe wyrażenie regularne (dla kwoty w zakresie 0.01 - 200000.00):  <math display="block">\wedge 0 \. (0) ([1-9]) \\$   \wedge 0 \. ( ([1-9]) (\d) ? ) \\$   \wedge ([1-9]) ( (\. \d {1,2} ) ? ) \\$   \wedge ( (? ! 0) (\d) {1,5} ) ( (\. \d {1,2} ) ? ) \\$   \wedge ( 1 (\d {5} ) ( \. \d {1,2} ) ? ) \\$   \wedge ( 200000 ( \. [0] {1,2} ) ? ) \\$</math>  Przykład:  amount = 42.82</p>

ciąg dalszy na następnej stronie

Table 1 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
currency	<p>Waluta określająca parametr amount , format zgodny ze standardem ISO 4217<sup>5</sup> .</p> <p>Dostępne wartości:            PLN, EUR, USD, GBP, JPY, CZK, SEK, UAH, RON, NOK, BGN, CHF, HRK, HUF, RUB</p> <p>Przykład:            currency = EUR</p>
description	<p>Opis przeprowadzanej operacji (transakcji).</p> <p>typ: <i>string</i>            minimalna długość: 1            maksymalna długość: 255</p> <p>Przykład:            description = Zamówienie nr 120/2018</p>
chk	<p>Suma kontrolna służąca do weryfikacji poprawności przesłanych danych. Opis funkcjonalności oraz sposób liczenia sumy znajduje się w rozdziale: <i>Ochrona integralności parametrów przekierowania (CHK)</i>.</p> <hr/> <p><b>Ważne:</b> Parametr domyślnie wymagany. Zalecana metoda liczenia sumy kontrolnej wymaga korzystania z odpowiedniej wersji api: <code>api_version = next</code></p> <hr/> <div style="border: 1px solid black; padding: 5px;"> <p><b>Ostrzeżenie:</b> Jeżeli konfiguracja sklepu wymusza przesłanie w żądaniu parametru <code>chk</code> (domyślne ustawienie sklepu po stronie systemu Dotpay) a jednak nie zostanie on wysłany, wówczas żądanie zostanie potraktowane jako błędnie. Tym samym zostanie ono odrzucone, czyli system odpowie błędem i nie będzie dalej przetwarzał takiego żądania. W logach dostępnych w panelu administracyjnym sprzedawcy Dotpay, w zakładce <i>Logi</i> dla takiego zdarzenia powinien pojawić się wpis o treści " <i>Didn't receive chk. Chk is required for api version next</i>"</p> </div>

### 2.3 Tabela 2. (Dodatkowe parametry przesyłane do serwisu Dotpay)

<sup>4</sup> [https://www.dotpay.pl/developer/doc/api\\_payment/v1.78.22.1/](https://www.dotpay.pl/developer/doc/api_payment/v1.78.22.1/)

<sup>5</sup> [https://pl.wikipedia.org/wiki/ISO\\_4217](https://pl.wikipedia.org/wiki/ISO_4217)

PARAMETR	ZNACZENIE / OPIS
channel	<p>Kanał płatności, jaki ma zostać zaznaczony po przeniesieniu klienta na strony serwisu Dotpay.</p> <p>Dostępne wyłącznie wartości liczbowe, które zostały przedstawione w sekcji ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI)</p> <p>Przykład: channel = 1</p> <p>Po przesłaniu wartości 1 zostanie zaznaczony kanał mTransfer.</p> <p>Lista dostępnych kanałów płatności dla wybranego sklepu (id) może być pobrana za pośrednictwem:</p> <ul style="list-style-type: none"> <li>• <a href="#">API panelu administracyjnego sprzedawcy</a><sup>6</sup></li> <li>• API płatności (mechanizm ten wykorzystywany jest również przez gotowy widжет służący do wyświetlenia dostępnych kanałów, jego opis znajduje się w panelu administracyjnym sprzedawcy, w zakładce: <i>Narzędzia -&gt; Widget</i> )</li> </ul> <p>Adres zasobu: <a href="https://ssl.dotpay.pl/t2/payment_api/channels/">https://ssl.dotpay.pl/t2/payment_api/channels/</a></p> <p>Wymagane parametry: id , amount , currency</p> <p>Opcjonalne parametry: lang , format (json lub xml)</p> <p>Przykład:  <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>Wymuszenie kanału podanego w parametrze channel .</p> <p>Przesłanie parametru ch_lock z wartością 1 nie pozwala kupującemu na wybranie innego kanału płatniczego, niż zdefiniowany przez serwis sprzedawcy w wykonanym przekierowaniu do serwisu Dotpay.</p> <p>Dostępne wartości:</p> <p>0 - Kanał nie będzie wymuszany (domyślnie).</p> <p>1 - Wymuszenie kanału przesłanego w parametrze channel .</p> <p>Przykład:</p> <p>Przesłanie w przekierowaniu odpowiednio parametrów channel = 1 oraz ch_lock = 1 uniemożliwi klientowi wybranie innego kanału płatności niż mTransfer.</p>
ignore_last_payment_channel	<p>Przesłanie parametru spowoduje zignorowanie ostatnio wybranej przez klienta metody płatności (zapamiętanej w pamięci jego przeglądarki), tzn. na stronie Dotpay wyświetli się pełna lista dostępnych kanałów, jak przy pierwszej płatności.</p> <p>Domyślnie podczas kolejnej płatności prezentowana jest ostatnio wybrana metoda, wraz z możliwością ewentualnej zmiany na inną.</p> <p>Dostępne wartości:</p> <p>1 - Zignorowanie ostatnio wybranej metody płatności</p> <p>Przykład:  ignore_last_payment_channel = 1</p>

ciąg dalszy na następnej stronie



Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
channel_groups	<p>Grupa kanałów płatności, jaka ma zostać wyświetlona po przeniesieniu klienta na strony serwisu Dotpay (domyślnie prezentowane są wszystkie).</p> <p>W przypadku korzystania z niniejszego parametru zalecane jest użycie ignore_last_payment_channel opisanego powyżej.</p> <p>Dostępne wartości:</p> <ul style="list-style-type: none"> <li>K - karty płatnicze,</li> <li>T - szybkie transfery,</li> <li>P - przelewy online,</li> <li>G - płatności gotówkowe,</li> <li>W - portmonetki i vouchery,</li> <li>R - raty,</li> <li>M - płatności mobilne (DCB),</li> <li>O - płatności odroczone,</li> <li>U - usługa inicjowania transakcji płatniczej (PIS: Payment Initiation Service),</li> <li>I - inne.</li> </ul> <p>W wartości parametru może być przekazanych kilka grup, w takiej sytuacji kolejne litery należy odseparować używając przecinka.</p> <p>Przykład:</p> <pre>channel_groups = T channel_groups = R,I channel_groups = R,I,P</pre>
url	<p>Adres internetowy (<i>HTTP</i> lub <i>HTTPS</i>) na jaki ma powrócić kupujący po dokonaniu płatności. Sterowanie zachowaniem parametru url określa parametr type .</p> <p>typ: <i>string</i> maksymalna długość: 1000</p> <p>Przykład:</p> <pre>url = https://www.example.com/thanks_page.php url = http://195.150.9.55/thanks_page.php</pre> <p>W przypadku przesłania parametrów url i type = 0 po zakończeniu procesu płatności kupującemu zostanie przedstawiony przycisk powrotu do serwisu sprzedawcy. Skorzystanie z przycisku spowoduje przeniesienie kupującego na adres podany w parametrze url wraz z parametrem status (przekazanym przez <i>POST</i> i <i>GET</i>) z wartością OK lub FAIL, który zawiera informację o ewentualnym wystąpieniu błędów na stronach płatności.</p> <hr/> <p><b>Ważne:</b> Parametr status informuje jedynie o przebiegu procesu płatności. Informacje o aktualnym statusie transakcji (new, completed, rejected, itp.) zawiera zmienna operation_status przesyłana na adres urlc .</p> <hr/> <p>Przykład:</p> <pre>url = https://www.example.com/thanks_page.php?status=OK</pre>

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
type	<p>Parametr określający metodę odwołania do serwisu sprzedawcy. Wartość parametru type ma wpływ na zachowanie parametru url .</p> <p>Dostępne wartości:</p> <p>0 - Po dokonaniu płatności kupującemu zostanie udostępniony przycisk powrotu do serwisu sprzedawcy,</p> <p>2 - Brak reakcji, bez przycisku (wartość domyślna).</p> <p>4 - Nastąpi bezpośrednie przekierowanie do dostawcy kanału płatności (np. Banku), jak również po dokonaniu płatności i wylogowaniu z serwisu dostawcy kanału, kupujący zostanie przekierowany bezpośrednio do serwisu sprzedawcy. Do pełnego działania funkcjonalność wymaga przesłania pełnego zestawu parametrów niezbędnych do płatności danym kanałem. W przypadku braku pełnego zestawu parametrów kupujący będzie musiał uzupełnić dane na stronie Dotpay, natomiast dalsza część procesu płatności zostanie zrealizowana bez zmian, tj. nastąpi automatyczny powrót od dostawcy kanału (tzw. <b>on-site / white label</b> )</p> <hr/> <p><b>Ważne:</b> W przypadku korzystania z mechanizmu bezpośredniego przekierowania do dostawcy kanału (type = 4) należy pamiętać o umieszczeniu pól akceptacji zgód regulaminu płatności Dotpay (parametr bylaw ) oraz przetwarzania przez PayPro S.A. danych osobowych płacącego dla potrzeb realizacji procesu płatności (parametr personal_data ).</p> <hr/> <p>Przykład:</p> <p>Po przesłaniu poniższego zestawu parametrów, kupującemu zostanie wyświetlony (po dokonaniu płatności) przycisk pozwalający przejść do strony <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>Po przesłaniu poniższego zestawu parametrów oraz pełnego zestawu parametrów wymaganych dla płatności danym kanałem, kupujący zostanie przekierowany bezpośrednio (z pominięciem interfejsu Dotpay) do dostawcy kanału (np. Banku), a następnie, po wylogowaniu z serwisu dostawcy kanału, zostanie przekierowany bezpośrednio na adres url . Wykorzystanie niniejszej wartości pozwala stworzyć przekierowanie o schemacie np. <i>Sklep -&gt; Bank -&gt; Sklep</i>:</p> <pre>type = 4 url = https://www.example.com/thanks_page.php bylaw = 1 personal_data = 1</pre>

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
buttontext	<p>Treść, która zostanie wyświetlona na przycisku powrotu do serwisu sprzedawcy. Domyślną wartością jest Powrót do sklepu.</p> <p>typ: <i>string</i>                      minimalna długość: 4                      maksymalna długość: 100</p> <p>Przykład:                      buttontext = Wróć do www.example.com</p>
bylaw	<p>Parametr informujący o zaakceptowaniu przez klienta regulaminu płatności oraz polityki cookies PayPro S.A.. W przypadku korzystania z niniejszego parametru system sprzedawcy powinien wyświetlić kupującemu pole wyboru z treścią analogiczną do poniższej..</p> <p>Akceptuję <a href="https://ssl.dotpay.pl/t2/cloudfs1/magellan_media/regulamin_platnosci">regulamin płatności</a> PayPro S.A..</p> <p>Dostępne wartości: 1 - akceptacja regulaminu płatności</p> <p>Przykład:                      bylaw = 1</p>
personal_data	<p>Parametr informujący o wyświetleniu klientowi informacji na temat przetwarzanie danych osobowych przez PayPro S.A..</p> <p>W przypadku korzystania z niniejszego parametru system sprzedawcy powinien wyświetlić treść analogiczną do poniższej:</p> <p>Przyjmuję do wiadomości, że w celu realizacji procesu płatności Administratorem moich danych osobowych jest PayPro S.A.. (KRS 0000347935), 60-198 Poznań (Polska), Pastelowa 8, +48616006170, <a href="mailto:bok@dotpay.pl">bok@dotpay.pl</a>, zobacz <a href="https://ssl.dotpay.pl/t2/cloudfs1/magellan_media/rodo">regulamin</a> pełną treść klauzuli informacyjnej</p> <p>Dostępne wartości: 1 - informacja o zapoznaniu się z klauzulą informacyjną</p> <p>Przykład:                      personal_data = 1</p>

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
urlc	<p>Adres internetowy (<i>HTTP</i> lub <i>HTTPS</i>) do odbioru parametrów potwierdzających realizację lub odmowę operacji (transakcji). Dokładny opis mechanizmu powiadomień <i>URLC</i> został zawarty w rozdz. <i>ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)</i>. Zalecane jest, aby podany adres kierował bezpośrednio do odbierającego pliku. Jeżeli po drodze nastąpi przekierowanie typu <i>301</i> lub <i>302</i> (które zezwalają na zmianę metody <i>HTTP</i>) otrzymana od Dotpay notyfikacja może być pusta.</p> <p>Ewentualnie można ustawić na serwerze przekierowanie typu <i>307 / 308</i>, które na takie zachowanie nie pozwalają. Przesłanie parametru nadpisuje domyślną konfigurację sklepu ( <i>id</i> ) ustawioną w panelu administracyjnym sprzedawcy, w zakładce <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i> .</p> <hr/> <p><b>Ważne:</b> Jeżeli w panelu nie zostało odblokowane przyjmowanie parametru <i>urlc</i> z zewnętrznych źródeł (odznaczona opcja <i>Blokuj zewnętrzne urlc</i> w menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>), to przesyłany parametr <i>urlc</i> jest ignorowany.</p> <hr/> <p><b>Ważne:</b> Jeżeli opcja <i>HTTPS verify</i> z zakładki <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i> jest włączona (domyślne ustawienie), przesłany adres musi korzystać z protokołu <i>HTTPS</i>.</p> <hr/> <p>typ: <i>string</i>  maksymalna długość: <i>1000</i>  Przykład:  urlc = https://www.example.com/urlc_receiver.php  urlc = http://195.150.9.55/urlc_receiver.php</p>
expiration_date	<p>Data przedawnienia żądania płatności, zgodnie ze strefą czasową Europa/Warszawa. W przypadku korzystania z niniejszego parametru zalecane jest podpisanie żądania realizowanego do systemu Dotpay. Opis realizacji podpisu zawarty został w rozdziale <i>Ochrona integralności parametrów przekierowania (CHK)</i> .</p> <p>Format: <i>YYYY-MM-DD hh:mm:ss</i>  Przykład:  expiration_date = 2019-06-01 12:06:37</p>
control	<p>Parametr pozwalający na przechowanie ciągu (np. numeru zamówienia ze sklepu sprzedawcy) o długości do 1000 znaków. Parametr w formie niezmienionej jest odsyłany do serwisu sprzedawcy w powiadomieniu <i>URLC</i>.</p> <p>typ: <i>string</i>  maksymalna długość: <i>1000</i>  Przykład:  control = ec4bf09d3dbe0cb71e6abc3ea44a7273</p>

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
firstname	<p>Imię osoby dokonującej płatność.                      typ: <i>string</i>                      minimalna długość: 1                      maksymalna długość: 50                      wyrażenie regularne:  <code>^\p{L}0-9\s\-\_]{1,50}\$</code></p> <p>Przykład:                      firstname = Jan</p>
lastname	<p>Nazwisko osoby dokonującej płatność.                      typ: <i>string</i>                      minimalna długość: 1                      maksymalna długość: 50                      wyrażenie regularne:  <code>^\p{L}0-9\s\-\_]{1,50}\$</code></p> <p>Przykład:                      lastname = Nowak</p>
email	<p>Adres e-mail osoby dokonującej płatność. Na ten adres zostanie wysłane potwierdzenie operacji płatności.                      typ: <i>string</i>                      maksymalna długość: 100</p> <p>Przykład:                      email = jan.nowak@example.com</p>
street	<p>Nazwa ulicy.                      typ: <i>string</i>                      minimalna długość: 1                      maksymalna długość: 100                      wyrażenie regularne:  <code>^\p{L}0-9\.\s\-\'_,\]{1,100}\$</code></p> <p>Przykład:                      street = Wielicka</p>
street_n1	<p>Numer budynku.                      typ: <i>string</i>                      minimalna długość: 1                      maksymalna długość: 30                      wyrażenie regularne:  <code>^\p{L}0-9\s\-\_\/]{1,30}\$</code></p> <p>Przykład:                      street_n1 = 4</p>
street_n2	<p>Numer mieszkania/lokalu.                      typ: <i>string</i>                      maksymalna długość: 30                      wyrażenie regularne:  <code>^\p{L}0-9\s\-\_]{0,30}\$</code></p> <p>Przykład:                      street_n2 = 18</p>

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
state	Dodatkowy parametr adresu lub stan/region. typ: <i>string</i> maksymalna długość: 50 Przykład: state = NY
addr3	Dodatkowy parametr adresu. typ: <i>string</i> maksymalna długość: 50
city	Nazwa miejscowości. typ: <i>string</i> minimalna długość: 1 maksymalna długość: 50 wyrażenie regularne: $\text{^\p{L}0-9\.\s\-\'\_}\{1,50\}\$$ Przykład: city = Kraków
postcode	Kod pocztowy. typ: <i>string</i> maksymalna długość: 20 wyrażenie regularne: $\text{^\d\w\s\-\}\{0,20\}\$$ Przykład: postcode = 30-552
phone	Numer telefonu. typ: <i>string</i> minimalna długość: 3 maksymalna długość: 20 wyrażenie regularne: $\text{^\+\s0-9\-\_}\{3,20\}\$$ Przykład: phone = +48 127654321
country	Nazwa kraju, z którego pochodzi osoba dokonująca płatność. Format zgodny ze standardem ISO 3166-1 <sup>7</sup> (alfa-2 lub alfa-3) lub tekstowa skrócona angielska nazwa kraju. typ: <i>string</i> maksymalna długość: 50 Przykład: country = PL country = POL country = Poland

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
lang	<p>Język prezentowanych stron i formularzy dokonywania płatności.</p> <p>Brak przesłania parametru spowoduje wyświetlenie stron w języku przeglądarki płatącego. Jeżeli dany język nie jest dostępny, zostanie użyty angielski.</p> <p>Dostępne wartości:</p> <ul style="list-style-type: none"> <li>cs - język czeski,</li> <li>de - język niemiecki,</li> <li>en - język angielski,</li> <li>es - język hiszpański,</li> <li>fr - język francuski,</li> <li>hu - język węgierski,</li> <li>it - język włoski,</li> <li>pl - język polski,</li> <li>ro - język rumuński,</li> <li>ru - język rosyjski,</li> <li>uk - język ukraiński</li> <li>lt - język litewski</li> <li>lv - język łotewski</li> <li>sk - język słowacki</li> </ul> <p>Przykład: lang = en</p>
customer	<p>Dane odbiorcy oraz dostawy. Parametr wymagany w przypadku niektórych kanałów płatności (np. dla kanału 95 - "PayPo"). Dane te powinny zostać zakodowane do formatu JSON<sup>8</sup> a następnie kodowane przy użyciu Base64<sup>9</sup>. Lista obsługiwanych danych dla tego parametru oraz sposób ich kodowania został zaprezentowany w Rozdziale <i>Obsługa danych dostawy oraz płatącego</i>.</p> <p>Wartość nie jest zwracana do sprzedawcy (np. w notyfikacjach URLC lub panelu administracyjnym).</p> <hr/> <p><b>Ważne:</b> Do obsługi tego parametru dane konto ( id ) musi posiadać wymuszoną weryfikację CHK po stronie Dotpay.</p> <hr/> <p>typ: <i>string</i> Przykład: customer = eyJyZWdpc3RlcmVhX3NpbmNlIjoiMjAxNy0xMi0zM</p>
deladdr	<p>Adres dostawy. Parametr pełni jedynie funkcję informacyjną dla systemu Dotpay.</p> <p>Wartość nie jest zwracana do sprzedawcy (np. w notyfikacjach URLC lub panelu administracyjnym).</p> <p>typ: <i>string</i> maksymalna długość: 500</p> <p>Przykład: deladdr = Punkt odbioru 3, Warszawa, ul. Ogonowa 14</p>

ciąg dalszy na następnej stronie

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
p_info	<p>Nazwa odbiorcy płatności, która zostanie wyświetlona klientowi na stronie płatności serwisu Dotpay. W przypadku nieprzesłania parametru wyświetlona zostanie domyślna nazwa sklepu widoczna w panelu administracyjnym Dotpay.</p> <p>Przesłanie parametru nadpisuje domyślną konfigurację sklepu ( id ) ustawioną w panelu administracyjnym sprzedawcy, w zakładce <i>Ustawienia -&gt; Konfiguracja sklepu -&gt; Edycja</i>.</p> <p>typ: <i>string</i> maksymalna długość: 300</p> <p>Przykład: p_info = Sklep example.com</p>
p_email	<p>Adres e-mail, który zostanie wyświetlony kupującemu w celu kontaktu ze sprzedawcą. Przesłanie parametru nadpisuje domyślny adres sklepu podany podczas rejestracji w serwisie Dotpay.</p> <p>Przesłanie parametru nadpisuje domyślną konfigurację sklepu ( id ) ustawioną w panelu administracyjnym sprzedawcy, w zakładce <i>Ustawienia -&gt; Konfiguracja sklepu -&gt; Edycja</i>.</p> <p>typ: <i>string</i> maksymalna długość: 100</p> <p>Przykład: p_email = biuro@example.com</p>

ciąg dalszy na następnej stronie



Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
pid	<p>Link płatniczy dla danego sklepu ( id )generowany za pośrednictwem Panelu administracyjnego Dotpay ( w zakładce <i>Narzędzia</i> -&gt; <i>Generator linków płatniczych</i> -&gt; <i>Generuj link</i>) lub za pośrednictwem <i>API panelu administracyjnego</i><sup>10</sup>.</p> <p>typ: <i>string</i> długość: 32</p> <p>Przykład: pid = rfhu4jb5ym657g3xluf4bbqfmbj6t17</p> <p>Parametr pid może występować albo samodzielnie w linku płatniczym, np: <a href="https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17">https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17</a></p> <p>albo z dodatkowymi parametrami: lang , ignore_last_payment_channel , np: <a href="https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&amp;lang=pl&amp;ignore_last_payment_channel=1">https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbj6t17&amp;lang=pl&amp;ignore_last_payment_channel=1</a></p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>Ostrzeżenie:</b> Domyślnie dla sklepu wymagane jest przesłanie również parametru chk . W przypadku generowania linków płatniczych z poziomu panelu administracyjnego, parametr chk dodawany jest automatycznie do linku. Natomiast, gdy link płatniczy generowany jest ręcznie z parametrów lub za pomocą API panelu administracyjnego, konieczne jest wygenerowanie poprawnej wartości chk oraz dołączenie jej do linku płatniczego.</p> </div>

Table 2 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
blik_code	<p>Kod BLIK, zatwierdzający płatność niniejszym kanałem. W standardowym procesie kod jest podawany przez klienta na stronie dostawcy płatności, po wybraniu kanału w serwisie Dotpay.</p> <p>Przekazanie parametru pozwala skrócić ścieżkę płatności, gdyż kod może być podawany przez klienta już na stronie sklepu sprzedawcy wraz z innymi danymi zamówienia.</p> <p>typ: <i>string</i>  minimalna długość: 6  maksymalna długość: 6  wyrażenie regularne:</p> <pre>^\d{6}\$</pre> <p>Przykład:  blik_code = 264230</p>
gp_token	<p>Zakodowany przy użyciu funkcji <b>Base64</b> token otrzymany z Google Pay po wdrożeniu przez Akceptanta <a href="#">Google Pay API<sup>11</sup></a>. Zawiera on zaszyfrowane dane tokenizowanej karty płatniczej służące do realizacji płatności.</p> <p>Przekazanie parametru pozwala skrócić ścieżkę płatności, ponieważ autentykacja danych kartowych płatniczej następuje już na stronie sklepu sprzedawcy.</p> <p>typ: <i>string</i></p> <p>Przykład:  gp_token = eyJzaWduYXR1cmUiOiJNRVFDZSUZDSm5MQWl1Rk50N...</p>
ap_token	<p>Zakodowany przy użyciu funkcji <b>Base64</b> token otrzymany z Apple Pay po wdrożeniu przez Akceptanta Apple Pay. Zawiera on zaszyfrowane dane tokenizowanej karty płatniczej służące do realizacji płatności.</p> <p>Przekazanie parametru pozwala skrócić ścieżkę płatności, ponieważ autentykacja danych kartowych płatniczej następuje już na stronie sklepu sprzedawcy/urzędzeniu płatniczego.</p> <p>typ: <i>string</i></p> <p>Przykład:  ap_token = eyJzacmUiOiJ ... h0UXAwM2RcIn0ifQ==</p>

### 3 ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)

W celu przekazywania do serwisu sprzedawcy informacji o dokonanej operacji (transakcji) został stworzony mechanizm powiadomień *URLC* (HTTP request, połączenie asynchroniczne, callback), które wysyłane są za pomocą metody *POST* niezależnie od działań kupującego.

Powiadomienia kierowane są na adres, jaki sprzedawca może określić w ustawieniach danego sklepu w panelu Dotpay po zalogowaniu (menu *Ustawienia* -> *Powiadomienia*)

<sup>6</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/pl/#lista-kanalow-platnosci-sklepu-id](https://www.dotpay.pl/developer/doc/api_seller/pl/#lista-kanalow-platnosci-sklepu-id)

<sup>7</sup> [https://pl.wikipedia.org/wiki/ISO\\_3166-1](https://pl.wikipedia.org/wiki/ISO_3166-1)

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

<sup>9</sup> <https://pl.wikipedia.org/wiki/Base64>

<sup>10</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/pl/#tworzenie-linku-platniczego](https://www.dotpay.pl/developer/doc/api_seller/pl/#tworzenie-linku-platniczego)

<sup>11</sup> <https://www.dotpay.pl/developer/doc/google-pay/pl/>

-> *Konfiguracja urlc -> Edycja*) lub na adres pobrany z parametru *urlc* przesłanego przez system sprzedawcy w przekierowaniu kupującego do płatności (o ile w powyżej wymienionym menu odblokowana została opcja przyjmowania parametru *urlc* ze źródeł zewnętrznych, tj. odznaczona opcja: *Blokuj zewnętrzne urlc*). Jeśli adres odbioru notyfikacji *URLC* **NIE** korzysta z szyfrowanej wersji protokołu *HTTP* (tj. *HTTPS*), należy pamiętać o dezaktywacji opcji *HTTPS verify* oraz *SSL certificate verify* w powyżej wspomnianym menu.

Unikalność transakcji w systemie kontrahenta **NIE** powinna być opierana tylko i wyłącznie na wartości parametru `control == control`. Jeżeli z jakiegoś powodu klient będzie kilkakrotnie przekierowywał się pomiędzy sklepem a Dotpay, bądź pomiędzy Dotpay a bankowością, istnieje możliwość dostarczenia np. 2 powiadomień z rozbieżnymi statusami transakcji dla tej samej wartości `control`. Transakcja powinna być zidentyfikowana biorąc również pod uwagę `operation_number` - identyfikator transakcji nadawany przez Dotpay.

**Ostrzeżenie:** Brak weryfikacji wartości parametrów: `amount == operation_original_amount`, `currency == operation_original_currency`, oraz `signature` po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe.

Treść oraz rezultat wysyłek notyfikacji *URLC* można znaleźć po zalogowaniu do panelu Dotpay wchodząc w szczegóły danej operacji (klikając na jej numer) w zakładce *Płatności -> Lista transakcji*. Z poziomu tego samego miejsca istnieje również możliwość ich manualnego ponowienia.

**Ostrzeżenie:** System Dotpay na poprawnie odebrane przez system sprzedawcy powiadomienie *URLC* oczekuje odpowiedzi

*OK*

(**wyłącznie dwie duże litery, nic więcej - kodowanie UTF8 bez BOM, kod odpowiedzi HTTP 200 (OK)**), która jest potwierdzeniem prawidłowo odebranego oraz przetworzonego powiadomienia, np.

```
echo \"OK\";
```

W przypadku zwrócenia przez system sprzedawcy odpowiedzi innej niż *OK*, system Dotpay przez pewien okres czasu będzie ponawiał (z częstotliwością ok. kilku, kilkunastu minut) wysyłanie powiadomienia. W nagłówku powiadomienia przesyłany może być parametr *X-Dotpay-URLC-Number* który dla każdego ponawianego powiadomienia będzie przyjmował wartość o jeden większą od poprzedniej, licząc od wartości 0 jako pierwszej wiadomości.

W poniższej tabeli przedstawione zostały parametry oraz opcjonalne wartości, jakie przesyłane są w powiadomieniach *URLC* kierowanych do systemu sprzedawcy.

---

**Informacja:** Parametry opcjonalne nie są domyślnie zwracane, aby je uzyskać należy wcześniej skontaktować się ze wsparciem technicznym Dotpay ([tech@dotpay.pl](mailto:tech@dotpay.pl)) w celu odpowiedniej konfiguracji konta.

---

---

**Informacja:** Domyślna konfiguracja systemu Dotpay wysyła notyfikacje *URLC* dla operacji typu *payment* dla statusów *completed* oraz *rejected* (wymienione w poniższej tabeli). Jeśli mają być przesyłane notyfikacje dla innych operacji oraz statusów, to fakt ten należy zgłosić na adres [tech@dotpay.pl](mailto:tech@dotpay.pl).

---

### 3.1 Tabela 3. (Parametry wysyłane przez serwis Dotpay po wykonaniu operacji (transakcji))

PARAMETR	ZNACZENIE / OPIS
id	ID sklepu sprzedawcy w systemie Dotpay, na rzecz którego wykonana została operacja (transakcja). typ: <i>integer</i> minimalna wartość: 1 maksymalna wartość: 999999 Przykład: id = 123456
operation_number	Numer operacji (transakcji). Format numeru opisuje wyrażenie regularne: $\text{\texttt{^\text{M}\d\{4,5}\ \ -\ \ \d\{4,5}\$}}$ Przykład: operation_number = M1234-56789
operation_type	Typ / rodzaj operacji. Dostępne wartości: payment - płatność, payment_multimerchant_child - płatność multimerchant, payment_multimerchant_parent - nadpłatność multimerchant, refund - zwrot, payout - wypłata, payout_any_amount - wypłata dowolnej kwoty, release_rollback - zwolnienie rollbacka, unidentified_payment - płatność niezidentyfikowana complaint - reklamacja credit_card_registration - rejestracja karty payout_commission - prowizja od wypłaty Przykład: operation_type = payment
operation_status	Status operacji (transakcji). Dostępne wartości: new - nowa, processing - oczekuje na wpłatę, completed - wykonana, rejected - odrzucona, processing_realization_waiting - oczekuje na realizację, processing_realization - realizowana Szczegółowe opisy statusów zostały przedstawione w rozdz. ZAŁĄCZNIK II (OPISY STATUSÓW OPERACJI)
	<b>Informacja:</b> Statusy completed i rejected są statusami końcowymi. Po ich osiągnięciu operacja nie zmieni statusu na inny.
	Przykład: operation_status = completed

ciąg dalszy na następnej stronie

Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
operation_amount	Kwota operacji zaksięgowanej w panelu Dotpay. Separatorem części setnej jest znak kropki. typ: <i>string</i> minimalna długość: 1 maksymalna długość: 10 Przykład: operation_amount = 177.27
operation_currency	Waluta określająca parametr operation_amount, format zgodny ze standardem standardem ISO 4217 <sup>12</sup> . Przykład: operation_currency = PLN
operation_withdrawal_amount	Parametr opcjonalny informujący o kwocie wypłaty środków z operacji. Przykład: operation_withdrawal_amount = 176.00
operation_commission_amount	Parametr opcjonalny informujący o pobranej prowizji. Parametr jest prezentowany jako kwota ujemna, dlatego zawiera znak -. Przykład: operation_commission_amount = -1.27
is_completed	Parametr opcjonalny informujący o oznaczeniu transakcji w panelu sprzedawcy, jako wykonanej. Przykład: is_completed = false is_completed = true
operation_original_amount	Kwota operacji (transakcji) pobrana z parametru amount jaki został przesłany przez serwis sprzedawcy w przekierowaniu Kupującego do serwisu Dotpay. typ: <i>string</i> minimalna długość: 1 maksymalna długość: 10 Przykład: operation_original_amount = 42.82 <hr/> <b>Ważne:</b> Brak weryfikacji wartości parametru operation_original_amount po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe. <hr/>

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
operation_original_currency	<p>Waluta operacji (transakcji) pobrana z parametru currency jaki został przesłany przez serwis sprzedawcy w przekierowaniu Kupującego do serwisu Dotpay, format zgodny ze standardem ISO 4217<sup>13</sup>.</p> <p>Przykład:  <code>operation_original_currency = EUR</code></p> <hr/> <p><b>Ważne:</b> Brak weryfikacji wartości parametru operation_original_currency po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe.</p> <hr/>
operation_datetime	<p>Data realizacji operacji (transakcji) lub zmiany statusu operacji.</p> <p>Format: <code>YYYY-MM-DD hh:mm:ss</code></p> <p>Przykład:  <code>operation_datetime = 2014-06-01 12:06:37</code></p>
operation_related_number	<p>Numer operacji (transakcji) powiązanej, jeśli takowa istnieje.</p> <p>Format numeru:  <code>^M\d{4,5}\-\d{4,5}\$</code></p> <p>Przykład:  <code>operation_related_number = M1234-56789</code></p> <p>Jeśli powiadomienie dotyczy zwrotu o numerze np. M9876-54321 wykonanego dla pierwotnej operacji M1234-56789, to parametr przyjmie wartość numeru pierwotnej operacji tj. M1234-56789.</p>
control	<p>Wartość odpowiada parametrowi jaki został przesłany przez serwis sprzedawcy w przekierowaniu do serwisu Dotpay podczas zlecenia płatności dla kupującego (parametr control w Tabela 2. <i>Dodatkowe parametry przesyłane do serwisu Dotpay</i>).</p> <p>typ: <i>string</i>  maksymalna długość: 1000</p> <p>Przykład:  <code>control = ec4bf09d3dbe0cb71e6abc3ea44a7273</code></p>
description	<p>Wartość odpowiada parametrowi jaki został przesłany przez serwis sprzedawcy w przekierowaniu do serwisu Dotpay podczas zlecenia płatności dla kupującego (parametr description w Tabela 1. <i>Podstawowe parametry przesyłane do serwisu Dotpay</i>).</p> <p>typ: <i>string</i>  minimalna długość: 1  maksymalna długość: 255</p> <p>Przykład:  <code>description = Faktura VAT 120/2014</code></p>
email	<p>Adres e mail podany przez osobę dokonującą płatność.</p> <p>typ: <i>string</i>  maksymalna długość: 100</p> <p>Przykład:  <code>email = jan.nowak@example.com</code></p>

ciąg dalszy na następnej stronie

Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
p_info	Nazwa odbiorcy płatności, która została wyświetlona klientowi na stronie płatności serwisu Dotpay podczas dokonywania płatności. typ: <i>string</i> maksymalna długość: 300 Przykład: p_info = Sklep komputerowy
p_email	Adres e-mail, który został wyświetlony kupującemu w celu kontaktu ze sprzedawcą. typ: <i>string</i> maksymalna długość: 100 Przykład: p_email = kontakt@sklep.pl
credit_card_issuer_identification_number	Parametr opcjonalny. Numer identyfikacyjny emitenta karty płatniczej, którą wykonana została płatność. <hr/> <b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i> ). <hr/> Przykład: credit_card_issuer_identification_number = 603753
credit_card_masked_number	Parametr opcjonalny. Zamaskowany numer karty płatniczej, którą wykonana została płatność. <hr/> <b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i> ). <hr/> Przykład: credit_card_masked_number = XXXX XXXX XXXX 6214
credit_card_expiration_year	Parametr opcjonalny. Rok daty ważności karty płatniczej, którą wykonana została płatność. <hr/> <b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i> ). <hr/> Przykład: credit_card_expiration_year = 2019

ciąg dalszy na następnej stronie

Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
credit_card_expiration_month	<p>Parametr opcjonalny. Miesiąc daty ważności karty płatniczej, którą wykonana została płatność.</p> <hr/> <p><b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>).</p> <hr/> <p>Przykład: credit_card_expiration_month = 01</p>
credit_card_brand_codename	<p>Parametr opcjonalny. Nazwa marki karty płatniczej, którą wykonana została płatność.</p> <hr/> <p><b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>).</p> <hr/> <p>Przykład: credit_card_brand_codename = visa</p>
credit_card_brand_code	<p>Parametr opcjonalny. Kod marki karty płatniczej, którą wykonana została płatność.</p> <hr/> <p><b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>).</p> <hr/> <p>Przykład: credit_card_brand_code = Visa</p>
credit_card_unique_identifier	<p>Parametr opcjonalny. Unikalny identyfikator karty zarejestrowanej w Dotpay.</p> <hr/> <p><b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>).</p> <hr/> <p>Przykład: credit_card_unique_identifier = 9b73chvjxofy4d9g...d78d7153ju34po12</p>
credit_card_id	<p>Parametr opcjonalny. Identyfikator karty nadany przez system Dotpay.</p> <hr/> <p><b>Ważne:</b> Do wysłania parametru wymagana jest aktywacja opcji <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> (menu <i>Ustawienia -&gt; Powiadomienia -&gt; Konfiguracja urlc -&gt; Edycja</i>). Dodatkowo dane konto ( id ) musi posiadać wymuszoną weryfikację <i>CHK</i> po stronie Dotpay.</p> <hr/> <p>Przykład: credit_card_id = 59f92e2bf8bedc36...1369dvpa4b7ab02ae</p>

ciąg dalszy na następnej stronie



Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
channel	Kanał płatności, jakim została wykonana operacja (transakcja). Dostępne wartości zostały przedstawione w rozdz. ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI) . Przykład: channel = 1
channel_country	Parametr opcjonalny. Informuje o kraju procesora płatności, poprzez którego została wykonana płatność. Format zgodny ze standardem ISO 3166-1 <sup>14</sup> (alfa-3). Przykład: channel_country = POL
geoip_country	Parametr opcjonalny. Informuje o lokalizacji kraju wynikającej z adresu IP z jakiego została wykonana płatność. Format zgodny ze standardem ISO 3166-1 <sup>15</sup> (alfa-3). Przykład: geoip_country = POL
payer_bank_account_name	Parametr opcjonalny. Nazwa właściciela rachunku zarejestrowana w banku. Przykład: payer_bank_account_name = JAN KOWALSKI
payer_bank_account	Parametr opcjonalny. Numer rachunku bankowego, z którego została wykonana płatność. Format zgodny ze standardem IBAN <sup>16</sup> . Przykład: payer_bank_account = PL41 1050 1009 4448 5481 1411 1395
payer_transfer_title	Parametr opcjonalny. Oryginalny tytuł operacji zarejestrowany w banku płatnika. Przykład: payer_transfer_title = Nazwa sklepu M5223-2008 Tytuł operacji Nr transakcji: M52232008000000000000000000000000
blik_voucher_pin	Parametr opcjonalny. PIN dla wydanego czeku BLIK. Przykład: blik_voucher_pin = 6365
blik_voucher_amount	Parametr opcjonalny. Wartość nominalna wydanego czeku BLIK. Przykład: blik_voucher_amount = 100.00

ciąg dalszy na następnej stronie

Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
blik_voucher_amount_used	<p>Parametr opcjonalny.</p> <p>Faktyczna kwota wypłacona z wydanego czeku BLIK. Kwota ta może być równa lub mniejsza wartości nominalnej wydanego czeku blik_voucher_amount.</p> <p>Przykład:</p> <p style="padding-left: 40px;">blik_voucher_amount_used = 60.00</p>
channel_reference_id	<p>Parametr opcjonalny.</p> <p>Dodatkowe szczegóły operacji, np. bankowy numer referencyjny.</p> <p>Przykład:</p> <p style="padding-left: 40px;">channel_reference_id = CDEd3pis6offk708bac2070ebf478183ad91668fd495a9884</p>
operation_seller_code	<p>Parametr opcjonalny (dostępny jedynie po wcześniejszych uzgodnieniach).</p> <p>Kod odpowiedzi dla transakcji odmownych, opisujący możliwy powód odmowy realizacji transakcji.</p> <p>Przykładowe kody prezentowane są w tabeli (odpowiednik parametru <i>seller_code</i> w api seller).</p> <p>Przykład:</p> <p style="padding-left: 40px;">operation_seller_code = CC_DO_NOT_HONOUR</p>

ciąg dalszy na następnej stronie

Table 3 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
signature	<p>Suma kontrolna będąca wynikiem działania funkcji skrótu <i>SHA-256</i> z konkatenacji powyższych parametrów według poniższego wzoru:</p> <p><i>PIN</i> + <i>id</i> + <i>operation_number</i> + <i>operation_type</i> + <i>operation_status</i> + <i>operation_amount</i> + <i>operation_currency</i> + <i>operation_withdrawal_amount</i> + <i>operation_commission_amount</i> + <i>is_completed</i> + <i>operation_original_amount</i> + <i>operation_original_currency</i> + <i>operation_datetime</i> + <i>operation_related_number</i> + <i>control</i> + <i>description</i> + <i>email</i> + <i>p_info</i> + <i>p_email</i> + <i>credit_card_issuer_identification_number</i> + <i>credit_card_masked_number</i> + <i>credit_card_expiration_year</i> + <i>credit_card_expiration_month</i> + <i>credit_card_brand_codename</i> + <i>credit_card_brand_code</i> + <i>credit_card_unique_identifier</i> + <i>credit_card_id</i> + <i>channel</i> + <i>channel_country</i> + <i>geop_country</i> + <i>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> + <i>channel_reference_id</i> + <i>operation_seller_code</i></p> <hr/> <p><b>Ważne:</b> Znak + w powyższym wzorze został użyty wyłącznie dla uzyskania czytelności. NIE jest wykorzystywany podczas wyliczania sumy kontrolnej. Jeśli w przesłanej notyfikacji nie jest obecny dany parametr, należy uznać go za wartość pustą lub pominąć w wyliczeniu.</p> <hr/> <p><b>Informacja:</b> <i>PIN</i> używany do wyliczania parametru <i>signature</i> to ciąg znaków, który sprzedawca musi wygenerować / określić dla danego sklepu ( <i>id</i> ) w panelu Dotpay po zalogowaniu (menu <i>Ustawienia</i> -&gt; <i>Powiadomienia</i> -&gt; <i>Konfiguracja urlc</i>).</p> <p>Służy on jedynie do wyliczenia sumy kontrolnej i <b>NIE powinien</b> być wysyłany w danych płatności!</p> <hr/> <p><b>Ważne:</b> Brak weryfikacji wartości parametru <i>signature</i> po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe.</p> <hr/>

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

<sup>13</sup> [https://pl.wikipedia.org/wiki/ISO\\_4217](https://pl.wikipedia.org/wiki/ISO_4217)

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

<sup>15</sup> [https://pl.wikipedia.org/wiki/ISO\\_3166-1](https://pl.wikipedia.org/wiki/ISO_3166-1)

<sup>16</sup> [http://pl.wikipedia.org/wiki/International\\_Bank\\_Account\\_Number](http://pl.wikipedia.org/wiki/International_Bank_Account_Number)

## 3.2 Przykład liczenia parametru signature

Poniżej został zamieszczony przykład wyliczenia wartości parametru signature w języku PHP.

```
1 <?php
2
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 DODATKOWE FUNKCJONALNOŚCI

### 4.1 Ochrona integralności parametrów przekierowania (CHK)

W celu zabezpieczenia integralności parametrów przekierowania realizowanego z serwisu sprzedawcy, system Dotpay umożliwia podpisywanie wartości przesyłanych parametrów.

Podpis powinien zostać przekazany, jako dodatkowy parametr `chk` (wraz z resztą parametrów żądania kierowanego do strony płatności systemu Dotpay).

**Ostrzeżenie:** Wszystkie parametry użyte w przekierowaniu do api Dotpay muszą mieć dokładnie taką nazwę jak opisuje niniejsza dokumentacja. Użycie alternatywnych nazw tych parametrów lub inna wielkość znaków w nazwie spowoduje niezgodność wyliczonej sumy kontrolnej.

Wartości użytych parametrów powinny być traktowane jako string i kodowane w UTF-8.

Jeżeli konfiguracja sklepu (*Ustawienia -> Konfiguracja sklepu -> Edycja*) wskazuje na wersję API inną niż `next`, to suma kontrolna liczona dla parametru `chk` zostanie potraktowana jako błędnie wyliczona - niezgodnie z niniejszą dokumentacją. Tym samym żądanie zostanie odrzucone, czyli system odpowie błędem i nie będzie dalej przetwarzał takiego żądania. W logach dostępnych w panelu administracyjnym sprzedawcy Dotpay, w zakładce *Logi* dla takiego zdarzenia powinien pojawić się wpis o treści „*Required Chk obtained, but hash not equal.*”

Nie dotyczy to sytuacji gdy w konfiguracji sklepu zezwolono na nadpisanie ustawień api w konfiguracji sklepu ( `id` ) poprzez przesłanie w żądaniu parametru `api_version = next`.

**Zalecamy ustawienie w konfiguracji sklepu** (*Ustawienia -> Konfiguracja sklepu -> Edycja*) **wersję API** `next` !

Sposób liczenia sumy kontrolnej opisuje poniższy przykład:

1. Przykładowe dane zebrane w procesie tworzenia płatności:

Listing 1: Przykładowa tablica z wysyłanymi do api Dotpay parametrami

```
Array
(
    [id] => 123456
    [amount] => 98.53
    [currency] => PLN
    [description] => Order123
    [url] => https://www.example.com/thanks_page.
↳php
    [type] => 0
)
```

2. Do tablicy parametrów dodajemy dodatkowy parametr o nazwie `paramsList`. Wartość tego parametru wyznacza się następująco:

- a. Przygotowujemy ciąg znaków składający się z posortowanej alfabetycznie listy nazw wszystkich parametrów przygotowanych w procesie tworzenia płatności
- b. Posortowane alfabetycznie nazwy parametrów oddzielamy średnikiem

Otrzymujemy zaktualizowaną tablicę:

Listing 2: Zaktualizowana tablica parametrów z dodatkowym parametrem *paramsList*

```
Array
(
    [amount] => 98.53
    [currency] => PLN
    [description] => Order123
    [id] => 123456
    [type] => 0
    [url] => https://www.example.com/
    ↪thanks_page.php
    [paramsList] => amount;currency;
    ↪description;id;type:url
)
```

**Ostrzeżenie:** Parametru *paramsList* nie należy dodawać osobno w żądaniu wysłanym wprost do api dotpay. Parametr *paramsList* dodawany jest tylko do tablicy budowanej przy tworzeniu samego parametru *chk* i nie występuje on oddzielnie.

3. Koniecznie ponownie sortujemy alfabetycznie listę nazw wszystkich parametrów występujących w nowej tablicy:

Listing 3: Posortowana alfabetycznie zaktualizowana tablica parametrów

```
Array
(
    [amount] => 98.53
    [currency] => PLN
    [description] => Order123
    [id] => 123456
    [paramsList] => amount;currency;description;id;
    ↪type:url
    [type] => 0
    [url] => https://www.example.com/thanks_page.php
)
```

4. Tak przygotowaną tablicę serializujemy do JSONa (kodowanie UTF-8), bez dodatkowych białych znaków. Wszystkie parametry występujące w tablicy traktujemy jako stringi. Zalecamy stosowanie escapowania znaków (np. **JSON\_UNESCAPED\_SLASHES** w PHP, nie jest konieczne użycie **JSON\_UNESCAPED\_UNICODE**).

Listing 4: Tablica danych w formacie JSON

```
{
  "amount": "98.53",
  "currency": "PLN",
  "description": "Order123",
  "id": "123456",
  "paramsList": "amount;currency;description;id;type:url",
  "type": "0",
  "url": "https://www.example.com/thanks_page.php"
}
```

5. Wyliczamy podpis za pomocą algorytmu `hmac`<sup>17</sup> z podaniem unikalnego klucza ( *PIN* ) do konta Dotpay (id) jako klucza, z funkcją skrótu sha256. Wynikiem jest ciąg zakodowany systemem heksadecymalnym, który należy umieścić jako wartość parametru `chk`

**Ostrzeżenie:** *PIN* służy jedynie do wyliczenia sumy kontrolnej i nie należy go podawać jako osobnego parametru w żądaniach wysyłanych do Dotpay!

Listing 5: Przykłady funkcji (w języku PHP) generujący wartość parametru `chk` z tablicy `$ParametersArray`

```
<?php
$DotpayPin = "P0lj9b2xIl87u1hCauuT4SFw6RmF01Tuy";

function GenerateChk($DotpayPin, $ParametersArray)
{
    //sorting the parameter list
    ksort($ParametersArray);

    // Display the semicolon separated list
    $paramList = implode(';', array_
↪keys($ParametersArray));

    //adding the parameter 'paramList' with
↪sorted list of parameters to the array
    $ParametersArray['paramsList'] =
↪$paramList;

    //re-sorting the parameter list
    ksort($ParametersArray);

    //json encoding with JSON_UNESCAPED_
↪SLASHES
    $json = json_encode(
↪$ParametersArray, JSON_UNESCAPED_SLASHES);

    // generate hash
    return hash_hmac('sha256', $json,
↪$DotpayPin, false);
}

// Source data array with parameters to be
↪redirected to api dotpay:
```

(ciąg dalszy na następnej stronie)

<sup>17</sup> <https://pl.wikipedia.org/wiki/HMAC>

(kontynuacja poprzedniej strony)

```
/* Important!
    All values should be
↳string.
    If you use a variable
↳here, put a declaration (string) before the value.
    Other values should
↳be enclosed in quotation marks, especially numerical ones.
*/

$ParametersArray = array(
    "id" => "123456",
    "amount" => "98.53",
    "currency" => "PLN",
    "description" =>
↳"Order123",
    "url" => "https://www.
↳example.com/thanks_page.php",
    "type" => "0"
);

//adding the chk parameter to the existing
↳original array
$ParametersArray['chk'] = GenerateChk($DotpayPin,
↳$ParametersArray);

//checking if everything works
print_r($ParametersArray);

?>
```

Listing 6: Wynik działania powyższego skryptu to zaktualizowana tablica parametrów z dodatkową sumą kontrolną chk

```
Array
(
    [id] => 123456
    [amount] => 98.53
    [currency] => PLN
    [description] => Order123
    [url] => https://www.example.com/thanks_page.php
    [type] => 0
    [chk] => 129db88a7f18bbb813a8c9c43a4bc5857fcb2d65d56c7f97dd77bd09d7e9ae7
)
```

W przypadku linków wygenerowanych za pomocą panelu administracyjnego ( pid ), wyliczanie wartości chk odbywa się z wykorzystaniem tylko parametru pid (zgodnie z powyższym wzorem), nie należy uwzględniać parametrów, z których powstał sam pid .

Listing 7: Posortowana tablica danych w formacie JSON dla pojedynczego parametru pid

```
{
    "paramsList": "pid",
    "pid": "Yhhu4jb5ym987g3xluf4bbqfmbj6t87"
}
```

Jeśli w przekierowaniu są wykorzystywane inne parametry oprócz pid, należy dodać je do wyliczenia, np:



Listing 8: Posortowana tablica danych w formacie JSON dla parametru pid oraz lang

```
{
  "lang": "en",
  "paramsList": "lang;pid",
  "pid": "rfhu4jb5ym657g3xluf4bbqfmbyj6t17"
}
```

---

**Informacja:** PIN używany do wyliczania parametru chk to ciąg znaków, który sprzedawca musi wygenerować / określić dla danego sklepu ( id ) w panelu Dotpay po zalogowaniu (menu *Ustawienia -> Powiadomienia -> Konfiguracja urlc*).

---

**Ostrzeżenie:** Domyślna konfiguracja sklepu ( id ) weryfikuje parametr chk przesłany w zleceniu płatności.

W przypadku nieprzesłania parametru lub jego niepoprawnej wartości zostanie zwrócony komunikat błędu.

Weryfikacja parametru chk nie jest jedynym zabezpieczeniem płatności. W niektórych modelach integracji może się okazać, że nie jest ona niezbędna i można z niej zrezygnować. Metoda weryfikacji chk sprawdza poprawność przesłanych parametrów na pierwszym etapie płatności: przekierowanie kupującego z serwisu Sprzedawcy do Dotpay. Najważniejszym etapem weryfikacji płatności jest potwierdzenie kwoty oraz waluty zaksięgowanej wpłaty w serwisie Dotpay za dane zamówienie z wartością zapisaną w serwisie Sprzedawcy, np. Rozdział *ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)*.

Jeśli Twój sklep nie ma włączonej wymagalności weryfikacji parametru chk, jest ona wyłączona lub opcjonalna, to poproś Dotpay o zmianę ustawień Twojego konta.

Jeżeli nie jesteś pewien czy Twoja integracja płatności z Dotpay obsługuje poprawnie parametr chk, skontaktuj się z Twoim dostawcą usługi / programistą odpowiedzialnym za integrację lub ze wsparciem technicznym Dotpay ( email: [tech@dotpay.pl](mailto:tech@dotpay.pl) ).

W przypadku gdy jednak chcesz zrezygnować z dodatkowego zabezpieczenia płatności i jesteś świadom ewentualnego ryzyka, które za tym stoi, możesz zgłosić chęć wyłączenia weryfikacji parametru chk dla swojego sklepu ( id ). Prośbę taka należy zgłosić na adres email: [administracja@dotpay.pl](mailto:administracja@dotpay.pl) .

Poniżej zostały zamieszczone przykłady funkcji (w języku PHP) generujące przekierowanie *POST / GET* wraz z wyliczeniem wartości parametru chk .

#### 4.1.1 Przykład 1 podstawowy

Dane zebrane przez sprzedawcę w procesie składania zamówienia przez płacącego należy umieścić w tablicy \$ParametersArray. Trzeba pamiętać by zadbać o poprawność nazw użytych parametrów (używać jedynie niezbędnych oraz opisanych w niniejszej dokumentacji). W tym przykładzie sprzedawca musi zadbać o poprawną walidację wartości przesyłanych danych tak by płacący wypełniający swoje dane na formularzu sprzedawcy nie używał zakazanych znaków w poszczególnych parametrach.

Pobierz poniższy przykład (po pobraniu zmień rozszerzenie pliku na .php): Pobierz przykład

```

1 <?php
2
3 ##### https://www.dotpay.pl/developer/doc/api_payment/ #####
4 ↪#####
5 #
6 # Exemplary function (PHP) generating the correct payment redirection (POST /
7 ↪GET) to Dotpay payment api with parametr 'chk' (checksum).
8 # You enter the payment data in the parameter: $ParametersArray.
9 #
10 # PayPro S.A.
11 # Tech Customer Service: tech@dotpay.pl
12 # Date: 2020-05-19
13 #
14 ↪#####
15
16 /** ----- BASE CONFIG ----- */
17
18 // Your Dotpay ID shop (6 digits)
19 $DotpayId = "123456";
20
21 // PIN for Your Dotpay ID (copy this from your dotpay panel carefully, without
22 ↪space)
23 $DotpayPin = "MyDotpayPIN000000j7yytSgMPXlg200";
24
25 // Dotpay Environment, available: "test" or "production"
26 $Environment = "test";
27
28 //Redirection method: POST or GET ; recommended method is "POST"
29 $RedirectionMethod = "POST";
30
31 // Auto submit form: available: true or false:
32 // If true - you do not need to do a click to the form. Forwarding to the API will
33 ↪be automatically
34 $autosubmit = true;
35
36 /** ----- end config ----- */
37
38 // ** ----- SAMPLE DATA ----- */
39
40 /* ## SAMPLE PAYMENT DATA IN ## */
41 // Note! You can use more parameters if You need
42 // You must give at least: 'amount', 'currency', 'description' (and of course ID
43 ↪and PIN in the configuration of this script)
44 // see more: https://www.dotpay.pl/developer/doc/api_payment/en/index.html#tabela-
45 ↪1-podstawowe-parametry-przesylane-do-serwisu-dotpay
46 // and: https://www.dotpay.pl/developer/doc/api_payment/en/index.html#tabela-2-
47 ↪dodatkowe-parametry-przesylane-do-serwisu-dotpay
48
49 /* Important!
50
51 All values should be string.
52 If you use a variable here, put a declaration (string) before the value.
53 Other values should be enclosed in quotation marks, especially numerical ones.
54 */
55 $ParametersArray = array(

```

(ciąg dalszy na następnej stronie)

```

55     "id" => (string) $DotpayId,
56     "api_version" => "next",
57     "amount" => "345.12",
58     "currency" => "PLN",
59     "description" => "Order no. 8765GfdC6",
60     "url" => "https://www.example.com/thanks_page.php",
61     "type" => "0",
62     "urlc" => "https://www.example.com/urlc_receiver.php",
63     "control" => "8765GfdC6-dsgfdg-2342235",
64     "firstname" => "Jan",
65     "lastname" => "Nowak",
66     "email" => "jan.nowak@example.com"
67 );
68
69 // ** ----- SAMPLE DATA end ----- **/
70
71
72 // if you do not know what configuration is on your account, add this parameter_
↳safely
73
74 if(!isset($ParametersArray['api_version']) && $ParametersArray['api_version'] ==
↳"next"){
75     $ParametersArray['api_version'] = "next";
76 }
77
78
79 ## function: counts the checksum from the defined array of all parameters
80
81 function GenerateChk($DotpayPin, $ParametersArray)
82 {
83
84     //sorting the parameter list
85     ksort($ParametersArray);
86
87     // Display the semicolon separated list
88     $paramList = implode(';', array_keys($ParametersArray));
89
90     //adding the parameter 'paramList' with sorted list of parameters to the array
91     $ParametersArray['paramsList'] = $paramList;
92
93     //re-sorting the parameter list
94     ksort($ParametersArray);
95
96     //json encoding
97     $json = json_encode($ParametersArray, JSON_UNESCAPED_SLASHES);
98
99
100     return hash_hmac('sha256', $json, $DotpayPin, false);
101 }
102
103
104 ## Function: Generate simple FORM to DOTPAY
105
106 function GenerateChkDotpayRedirection($DotpayPin, $Environment, $RedirectionMethod,
↳ $ParametersArray, $next_chk, $autosubmit)
107 {
108
109
110     if ($Environment == 'production') {
111         $EnvironmentAddress = 'https://ssl.dotpay.pl/t2/';
112     } elseif ($Environment == 'test') {

```

```

113     $EnvironmentAddress = 'https://ssl.dotpay.pl/test_payment/';
114 }
115
116
117
118     if ($RedirectionMethod == 'POST') {
119         $RedirectionCode = '<form action="' . $EnvironmentAddress . '" method="POST
↳ " id="dotpay_redirection_form" accept-charset="UTF-8">' . PHP_EOL;
120
121         foreach ($ParametersArray as $key => $value) {
122             $RedirectionCode .= "\t" . '<input name="' . $key . '" value="' .
↳ $value . '" type="hidden"/>' . PHP_EOL;
123         }
124         $RedirectionCode .= "\t" . '<input name="chk" value="' . $next_chk . '"
↳ type="hidden"/>' . PHP_EOL;
125         $RedirectionCode .= '</form>' . PHP_EOL . '<button id="dotpay_redirection_
↳ button" type="submit" form="dotpay_redirection_form" value="Submit">Confirm and
↳ Pay</button>' . PHP_EOL;
126
127         //auto submit form
128         if ($autosubmit == true) {
129             $RedirectionCode .= "<script type=\"text/javascript\">
↳ setTimeout(function(){document.getElementById('dotpay_redirection_form').
↳ submit();}, 10);</script>";
130         }
131
132         return $RedirectionCode;
133
134     } elseif ($RedirectionMethod == 'GET') {
135         $RedirectionCode = $EnvironmentAddress . '?';
136
137         foreach ($ParametersArray as $key => $value) {
138             $RedirectionCode .= $key . '=' . rawurlencode($value) . '&';
139         }
140
141         $RedirectionCode .= 'chk=' . $next_chk;
142
143         return '<a href="' . $RedirectionCode . '">Link to Pay</a>';
144     } else {
145         return 'configuration error';
146
147     }
148 }
149
150
151 #####
152
153 // Calculate checksum for 'chk' parameter:
154
155 $next_chk = GenerateChk($DotpayPin, $ParametersArray);
156
157
158 /*
159     Print the form according to the settings:
160     get form (POST method) or payment link (GET method)
161     ("account PIN", "[test|production]", "[POST|GET]", "payment data", "chk_value",
↳ "[true|false]")
162
163 */
164
165 echo GenerateChkDotpayRedirection($DotpayPin, $Environment, $RedirectionMethod,
↳ $ParametersArray, $next_chk, $autosubmit);

```

(ciąg dalszy na następnej stronie)

166  
167

?&gt;

#### 4.1.2 Przykład 2 rozbudowany

Skrypt zawiera mechanizm walidacji oraz „czyszczenia” niektórych danych użytych w poszczególnych parametrach.

Dane zebrane przez sprzedawcę w procesie składania zamówienia przez płacącego należy przypisać do utworzonych zmiennych (np. \$order\_amount, \$payer\_firstname, \$payer\_lastname, \$dp\_description, itp.). Następnie zmienne te są później wykorzystywane w tablicy \$ParametersArray po wcześniejszym odfiltrowaniu. Walidacji dotyczą dane płacącego jak firstname, lastname, oraz dane dotyczące adresu płacącego gdzie najczęściej może dochodzić do błędnego wypełniania formularzy przez płacących z niedozwolonymi znakami w poszczególnych polach. Odpowiednie funkcje (np. CheckPaymentLang(), CheckPaymentCurrency(), CheckStreet(), CheckStreetN1(), CheckStreetN2(), CheckCountry(), CheckPostcode(), CheckCity(), CheckPhone(), CheckLastname(), CheckFirstname()) usuwają z takiego stringa niedozwolone znaki. W przypadku przekazania do zmiennej wartości dla currency innej niż dozwolona (funkcja getAcceptCurrency()) - skrypt nie wygeneruje formularza tylko zwróci wyjątek. W przypadku przekazania do zmiennej wartości dla lang innej niż dozwolona (funkcja getAcceptLang()) - skrypt ustawi wartość tego parametru na en (język angielski).

Analogicznie jeśli wśród parametrów ma być użyty customer - można również użyć jak wyżej podobnego rozwiązania przypisując dane do zmiennych a następnie do tablicy \$customer.

Trzeba pamiętać by zadbać o poprawność nazw użytych parametrów (używać jedynie niezbędnych oraz opisanych w niniejszej dokumentacji).

Pobierz poniższy przykład (po pobraniu zmień rozszerzenie pliku na .php): Pobierz przykład

```

1  <?php
2
3  ##### https://www.dotpay.pl/developer/doc/api_payment/ #####
4  ↪#####
5  #
6  #   Exemplary function (PHP) generating the correct payment redirection (POST /
7  ↪GET) to Dotpay payment api with parametr 'chk' (checksum).
8  #   You enter the payment data in the parameter: $ParametersArray (and $customer
9  ↪if you need)
10 #   Added some functions that prevent certain parameters from passing invalid
11 ↪characters to api dotpay
12 #
13 #   PayPro S.A.
14 #   Tech Customer Service: tech@dotpay.pl
15 #   Date: 2020-05-19
16 #
17 #####
18 ↪#####
19
20 /** ----- BASE CONFIG ----- */
21
22 // Your Dotpay ID shop (6 digits)
23 $DotpayId = "123456";
24
25 // PIN for Your Dotpay ID (copy this from your dotpay panel carefully, without
26 ↪space)

```

(ciąg dalszy na następnej stronie)

```

22 $DotpayPin = "MyDotpayPIN000000j4suuSgMPXlg100";
23
24 // Dotpay Environment, available: "test" or "production"
25 $Environment = "test";
26
27 //Redirection method: POST or GET ; recommended method is "POST"
28 $RedirectionMethod = "POST";
29
30 // Auto submit form: available: true or false (only if $RedirectionMethod = "POST")
31 // If true - you do not need to do a click to the form. Forwarding to the API will
   ↳be automatically
32 $autosubmit = false;
33
34 ## -----
   ↳-----
35
36 /** ----- ORDER CONFIG ----- */
37
38 /* ## sample order details for payment, used in array: $ParametersArray ## */
39
40 $order_amount = "4100.20";
41 $order_currency = "PLN";
42 $dp_description = "Order no. 003376483/33";
43 $dp_url = "https://www.example.com/thanks_page.php";
44 $dp_type = "0";
45 $dp_urlc = "https://www.example.com/urlc_receiver.php";
46 $dp_control = "KI87dLQWR3UIYT";
47 $payer_firstname = "Daniel";
48 $payer_lastname = "Nowak";
49 $payer_email = "daniel.nowak@example.com";
50 $payer_street = "Zielona";
51 $payer_street_n1 = "9a"; // Building number
52 $payer_street_n2 = "11"; // Flat number
53 $payer_city = "Pcim";
54 $payer_postcode = "00-345";
55 $payer_phone = "123456789";
56 $payer_country = "POL";
57 $payer_lang = "pl";
58 $dp_last_channel = "1";
59
60 /*
61 ... and other possible parameters if you need according to the
   ↳documentation:
62 https://www.dotpay.pl/developer/doc/api_payment/pl/index.html#tabela-2-
   ↳dodatkowe-parametry-przesylane-do-serwisu-dotpay
63
64 */
65
66
67
68 /*
69 ### SAMPLE CUSTOMER DATA IN with delivery address (optional), used in array:
   ↳$customer ### You can remove it if You don't need it
70 manual: https://www.dotpay.pl/developer/doc/api_payment/pl/index.html#obsługa-
   ↳danych-dostawy-oraz-placacego
71 */
72
73 $payer_first_name = "Adam";
74 $payer_last_name = "Kowal";
75 $payer_email = "mymail@example.com";
76 $customer_city = "Warszawa";

```

```

77 $customer_street = "Niebieska";
78 $customer_building = "52";
79 $customer_postcode = "00-953";
80
81
82
83 /** ----- end config ----- */
84
85
86 ### Below are some functions that prevent certain parameters from passing invalid_
↳characters to api dotpay ###
87
88 /**
89  * checks and crops the size of a string
90  * the $special parameter means an estimate of how many urlencode characters can_
↳be used in a given field
91  * e.g. 'ż' (1 char) -> '%C5%BC' (6 chars)
92  * replacing removing double or more special characters that appear side by side_
↳by space from: firstname, lastname, city, street, p_info...
93 */
94 function encoded_substrParams($string, $from, $to, $special = 0)
95 {
96     $string2 = preg_replace('/(\s{2,}|\.{2,}|\@{2,}|\-{2,}|\/{3,} | \'{2,}|\\"{2,})|_
↳{2,})/', ' ', $string);
97     $s = html_entity_decode($string2, ENT_QUOTES, 'UTF-8');
98     $sub = mb_substr($s, $from, $to, 'UTF-8');
99     $sum = strlen(urlencode($sub));
100     if ($sum > $to) {
101         $newsize = $to - $special;
102         $sub = mb_substr($s, $from, $newsize, 'UTF-8');
103     }
104     return trim($sub);
105 }
106
107 /**
108  * check, remove unnecessary characters and return customer firstname
109  * @return string
110 */
111 function CheckFirstname($firstName)
112 {
113     //allowed only: letters, digits, spaces, symbols _-.,'
114     $firstName = preg_replace('/[^\w _-]/u', '', $firstName);
115     $firstName1 = html_entity_decode($firstName, ENT_QUOTES, 'UTF-8');
116
117
118     $NewPersonName1 = preg_replace('/[^\p{L}0-9\s\_-]/u', '', $firstName1);
119     return encoded_substrParams($NewPersonName1, 0, 49, 24);
120 }
121
122 /**
123  * check, remove unnecessary characters and return customer lastname
124  * @return string
125 */
126 function CheckLastname($lastName)
127 {
128
129     //allowed only: letters, digits, spaces, symbols _-.,'
130     $lastName = preg_replace('/[^\w _-]/u', '', $lastName);
131     $lastName1 = html_entity_decode($lastName, ENT_QUOTES, 'UTF-8');
132
133     $NewPersonName2 = preg_replace('/[^\p{L}0-9\s\_-]/u', '', $lastName1);

```

(ciąg dalszy na następnej stronie)

```

134     return encoded_substrParams($NewPersonName2, 0, 49, 24);
135 }
136
137 /**
138  * check, remove unnecessary characters and return customer phone
139  * @return string
140  */
141 function CheckPhone($phone)
142 {
143     $phone = str_replace(' ', '', $phone);
144     $phone = str_replace('+', '', $phone);
145
146     $NewPhone1 = preg_replace('/[^\+\s0-9\-\_]/', '', $phone);
147     return encoded_substrParams($NewPhone1, 0, 19, 6);
148 }
149
150 /**
151  * check, remove unnecessary characters and return customer city
152  * @return string
153  */
154 function CheckCity($city)
155 {
156     //allowed only: letters, digits, spaces, symbols _-.,'
157     $city = preg_replace('/[^\.\w \'_-]/u', '', $city);
158     $city1 = html_entity_decode($city, ENT_QUOTES, 'UTF-8');
159
160     return encoded_substrParams($city1, 0, 49, 24);
161 }
162
163
164 /**
165  * check, remove unnecessary characters and return customer postcode
166  * @return string
167  */
168 function CheckPostcode($postcode, $country = null)
169 {
170
171     if (empty($postcode)) {
172         return $postcode;
173     }
174     if (preg_match('/^\d{2}\-\d{3}$/', $postcode) == 0 && strtolower($country ==
175     ↪'pl')) {
176         $postcode = str_replace('-', '', $postcode);
177         $postcode = substr($postcode, 0, 2) . '-' . substr($postcode, 2, 3);
178     }
179
180     $NewPostcode1 = preg_replace('/[^\d\w\s\-\_]/', '', $postcode);
181     return encoded_substrParams($NewPostcode1, 0, 19, 6);
182 }
183
184
185 /**
186  * check, remove unnecessary characters and return customer country
187  * @return string
188  */
189 function CheckCountry($country)
190 {
191
192     if (preg_match('/^[a-zA-Z]{2,3}$/', trim($country)) == 0) {
193         $country_check = null;

```



```

194     } else {
195         $country_check = trim($country);
196     }
197
198     return strtoupper($country_check);
199 }
200
201
202 /**
203  * check, remove unnecessary characters and return customer street
204  * @return string
205  */
206 function CheckStreet($street)
207 {
208
209     //allowed only: letters, digits, spaces, symbols _-.,'
210     $street = preg_replace('/[^\w \'-_]/u', '', $street);
211     $street1 = html_entity_decode($street, ENT_QUOTES, 'UTF-8');
212
213     return encoded_substrParams($street1, 0, 99, 50);
214 }
215
216
217 /**
218  * check, remove unnecessary characters and return customer street_n1 - building_
219  * ↪number
220  * @return string
221  */
222 function CheckStreetN1($street_n1)
223 {
224
225     //allowed only: letters, digits, spaces, symbols _-.,'
226     $street_n1 = preg_replace('/[^\p{L}0-9\s\-\_\/]/u', '', $street_n1);
227     $street1_n1 = html_entity_decode($street_n1, ENT_QUOTES, 'UTF-8');
228
229     return encoded_substrParams($street1_n1, 0, 29, 24);
230 }
231
232
233 /**
234  * check, remove unnecessary characters and return customer street_n2 - flat_
235  * ↪number.
236  * @return string
237  */
238 function CheckStreetN2($street_n2)
239 {
240
241     //allowed only: letters, digits, spaces, symbols _-.,'
242     $street_n2 = preg_replace('/[^\p{L}0-9\s\-\_]/u', '', $street_n2);
243     $street1_n2 = html_entity_decode($street_n2, ENT_QUOTES, 'UTF-8');
244
245     return encoded_substrParams($street1_n2, 0, 29, 24);
246 }
247
248 /**
249  * Return array of languages that are accepted by Dotpay
250  * @return array
251  */
252 function getAcceptLang()
253 {
254     return array(

```

```

253     'pl',
254     'en',
255     'de',
256     'it',
257     'fr',
258     'es',
259     'cz',
260     'cs',
261     'ru',
262     'hu',
263     'ro',
264     'uk',
265     'lt',
266     'lv',
267     'sk'
268 );
269 }
270
271
272 /**
273  * Return array of Curriences that are accepted by Dotpay
274  * @return array
275  */
276 function getAcceptCurrency()
277 {
278     return array(
279         'EUR',
280         'USD',
281         'GBP',
282         'JPY',
283         'CZK',
284         'SEK',
285         'UAH',
286         'RON',
287         'PLN',
288         'NOK',
289         'BGN',
290         'CHF',
291         'HRK',
292         'HUF',
293         'RUB'
294     );
295 }
296
297
298
299 /**
300  * Return payment language name
301  * @return string
302  */
303 function CheckPaymentLang($language)
304 {
305     $f_dotpay_lang = '';
306
307     if (is_string($language)) {
308         $languageArray = explode('-', $language);
309         if (isset($languageArray[0])) {
310             $languageLower = strtolower($languageArray[0]);
311             $f_dotpay_lang = $languageLower;
312         }
313     }

```

```

314
315     if ($f_dotpay_lang == 'pl') {
316         $dotpay_lang = 'pl';
317     } elseif (!in_array($languageLower, getAcceptLang())) {
318         $dotpay_lang = 'en';
319     } else {
320         $dotpay_lang = $languageLower;
321     }
322
323     return $dotpay_lang;
324 }
325
326
327 /**
328  * Check a currency code by comparing allowed ( getAcceptCurrency() function)
329  * @param string $currency Currency code
330  * return false when the given currency code is incorrect
331  */
332 function CheckPaymentCurrency($currency)
333 {
334     $currency = strtoupper($currency);
335
336     if (!in_array($currency, getAcceptCurrency())) {
337         $dotpay_currency = false;
338     } else {
339         $dotpay_currency = (string) $currency;
340     }
341
342     return $dotpay_currency;
343 }
344
345
346
347 /**
348  * Convert original amount using a dot as a decimal place regardless of the locale.
349  * @param float $amount
350  * @return string
351  *
352  */
353
354 function normalizeDecimalAmount($val)
355 {
356
357     $input = str_replace(' ', '', $val);
358     $number = str_replace(',', '.', $input);
359     if (strpos($number, '.') {
360         $groups = explode('.', str_replace(',', '.', $number));
361         $lastGroup = array_pop($groups);
362         $number = implode('.', $groups) . '.' . $lastGroup;
363     }
364     return bcadd($number, 0, 2);
365 }
366
367
368 // ** ----- SAMPLE DATA ----- **/
369
370 /* ## SAMPLE PAYMENT DATA IN ##
371
372 Note! You can use more parameters if You need. Case sensitive in parameter names
373 ↪ is important !
374 You must give at least: 'amount', 'currency', 'description' (and of course ID and
375 ↪ PIN in the configuration of this script)

```

(ciąg dalszy na następnej stronie)

```

374 see more: https://www.dotpay.pl/developer/doc/api\_payment/en/index.html#tabela-1-
↳podstawowe-parametry-przesylane-do-serwisu-dotpay
375 and: https://www.dotpay.pl/developer/doc/api\_payment/en/index.html#tabela-2-
↳dodatkowe-parametry-przesylane-do-serwisu-dotpay
376
377 Filters were used to remove forbidden characters entered e.g. in the address by
↳the payer from specific parameters.
378
379 */
380
381 /* Important!
382
383 All values should be string.
384 If you use a variable here, put a declaration (string) before the value.
385 Other values should be enclosed in quotation marks, especially numerical ones.
386 */
387
388 $ParametersArray = array(
389
390     "id" => (string) $DotpayId,
391     "api_version" => "next", // !important
392     "amount" => (string) normalizeDecimalAmount($order_amount),
393     "currency" => (string) CheckPaymentCurrency($order_currency),
394     "description" => (string) $dp_description,
395     "url" => (string) $dp_url,
396     "type" => (string) $dp_type,
397     "urlc" => (string) $dp_urlc,
398     "control" => (string) $dp_control,
399     "firstname" => (string) CheckFirstname($payer_firstname),
400     "lastname" => (string) CheckLastname($payer_lastname),
401     "email" => (string) $payer_email,
402     "street" => (string) CheckStreet($payer_street),
403     "street_n1" => (string) CheckStreetN1($payer_street_n1),
404     "street_n2" => (string) CheckStreetN2($payer_street_n2),
405     "city" => (string) CheckCity($payer_city),
406     "postcode" => (string) CheckPostcode($payer_postcode, strtolower($payer_lang)),
407     "phone" => (string) CheckPhone($payer_phone),
408     "country" => (string) CheckCountry($payer_country),
409     "lang" => (string) CheckPaymentLang($payer_lang),
410     "ignore_last_payment_channel" => (string) $dp_last_channel
411 );
412
413
414
415
416 /* ### SAMPLE CUSTOMER DATA IN with delivery address (optional) ###
417 You can remove it if You don't need it
418 */
419
420 // -----
421 $customer = array(
422
423     "payer" => array(
424         "first_name" => CheckFirstname($payer_first_name),
425         "last_name" => CheckLastname($payer_last_name),
426         "email" => $payer_email
427     ),
428     "order" => array(
429         "delivery_address" => array(
430             "city" => CheckCity($customer_city),
431             "street" => (string) CheckStreet($customer_street),

```

(ciąg dalszy na następnej stronie)

```

432         "building_number" => (string) CheckStreetN1($customer_building),
433         "postcode" => (string) CheckPostcode($customer_postcode)
434     )
435 )
436 );
437
438
439
440 if (empty($customer) || !isset($customer['payer']['first_name']) || !isset(
↪ $customer['payer']['last_name']) || !isset($customer['payer']['email']) || !
↪ isset($customer['order']['delivery_address']['city']) || !isset($customer['order
↪ ']['delivery_address']['street']) || !isset($customer['order']['delivery_address
↪ ']['building_number']) || !isset($customer['order']['delivery_address']['postcode
↪ '])) {
441     $customer_base64 = null;
442 } else {
443     $customer_base64 = base64_encode(json_encode($customer, JSON_UNESCAPED_UNICODE_
↪ | JSON_UNESCAPED_SLASHES));
444 }
445
446
447 if ($customer_base64 != null) {
448     $ParametersArray["customer"] = $customer_base64;
449 }
450
451
452
453 // ** ----- SAMPLE DATA end ----- **/
454
455
456 // if you do not know what configuration is on your account, add this parameter_
↪ safely
457
458 if(!(isset($ParametersArray['api_version']) && $ParametersArray['api_version'] ==
↪ "next")){
459     $ParametersArray['api_version'] = "next";
460 }
461
462
463
464 ## function: counts the checksum from the defined array of all parameters
465
466 function GenerateChk($DotpayPin, $ParametersArray)
467 {
468     //sorting the parameter list
469     ksort($ParametersArray);
470
471     // Display the semicolon separated list
472     $paramList = implode(';', array_keys($ParametersArray));
473
474     //adding the parameter 'paramList' with sorted list of parameters to the array
475     $ParametersArray['paramsList'] = $paramList;
476
477     //re-sorting the parameter list
478     ksort($ParametersArray);
479
480     //json encoding
481     $json = json_encode($ParametersArray, JSON_UNESCAPED_SLASHES);
482
483     return hash_hmac('sha256', $json, $DotpayPin, false);
484

```

```

485 }
486
487
488
489
490 ## Function: Generate simple FORM to DOTPAY
491
492 function GenerateChkDotpayRedirection($DotpayPin, $Environment, $RedirectionMethod,
↳ $ParametersArray, $next_chk, $autosubmit)
493 {
494
495     //$ParametersArray = array_change_key_case($ParametersArray, CASE_LOWER);
496
497
498     if ($Environment == 'production') {
499         $EnvironmentAddress = 'https://ssl.dotpay.pl/t2/';
500     } elseif ($Environment == 'test') {
501         $EnvironmentAddress = 'https://ssl.dotpay.pl/test_payment/';
502     }
503
504
505
506     if ($RedirectionMethod == 'POST') {
507         $RedirectionCode = '<form action="' . $EnvironmentAddress . '" method="POST'
↳ " id="dotpay_redirection_form" accept-charset="UTF-8">' . PHP_EOL;
508
509         foreach ($ParametersArray as $key => $value) {
510             $RedirectionCode .= "\t" . '<input name="' . $key . '" value="' .
↳ $value . '" type="hidden"/>' . PHP_EOL;
511         }
512         $RedirectionCode .= "\t" . '<input name="chk" value="' . $next_chk . '"
↳ type="hidden"/>' . PHP_EOL;
513         $RedirectionCode .= '</form>' . PHP_EOL . '<button id="dotpay_redirection_
↳ button" type="submit" form="dotpay_redirection_form" value="Submit">Confirm and
↳ Pay</button>' . PHP_EOL;
514
515         //auto submit form
516         if ($autosubmit == true) {
517             $RedirectionCode .= "<script type=\"text/javascript\">
↳ setTimeout(function(){document.getElementById('dotpay_redirection_form')
↳ submit();}, 10);</script>";
518         }
519
520         return $RedirectionCode;
521
522     } elseif ($RedirectionMethod == 'GET') {
523         $RedirectionCode = $EnvironmentAddress . '?';
524
525         foreach ($ParametersArray as $key => $value) {
526             $RedirectionCode .= $key . '=' . rawurlencode($value) . '&';
527         }
528
529         $RedirectionCode .= 'chk=' . $next_chk;
530
531         return '<a href="' . $RedirectionCode . '">Link to Pay</a>';
532     } else {
533         return 'configuration error';
534
535     }
536
537

```

```

538 }
539
540 #####
541
542 // Calculate checksum for 'chk' parameter:
543
544 $next_chk = GenerateChk($DotpayPin, $ParametersArray);
545
546
547
548 /*
549  Print the form according to the settings:
550  get form (POST method) or payment link (GET method)
551  ("account PIN", "[test|production]", "[POST|GET]", "payment data", "chk_value",
552  ↪ "[true|false]")
553 */
554 if (CheckPaymentCurrency($order_currency) != false) {
555
556     echo GenerateChkDotpayRedirection($DotpayPin, $Environment, $RedirectionMethod,
557     ↪ $ParametersArray, $next_chk, $autosubmit);
558
559 } else {
560     echo "The currency of the payment you want to use (" . $order_currency . ") is_
561     ↪ not allowed in the Dotpay system!";
562 }
563
564 ?>

```

## 4.2 One-click i płatności cykliczne

---

**Informacja:** Funkcjonalność dostępna jest wyłącznie dla sklepów ( id ), które zostały odpowiednio skonfigurowane po stronie systemu Dotpay, co zależy od warunków wynikających z podpisanych przez akceptantów umów.

---

Funkcjonalność dostępna jest dla kanału kart płatniczych (numer 248) i pozwala klientowi na realizację wpłat bez konieczności podawania pełnych danych karty podczas kolejnych płatności dokonywanych przez system Dotpay.

W celu skorzystania z funkcjonalności system sprzedawcy powinien przekazać w zleceniu płatności dodatkowe parametry (opisane poniżej), natomiast klient podczas pierwszego użycia danych konkretnej karty powinien wyrazić zgodę na ich zapisanie przez system Dotpay (dane przechowywane są zgodnie z najwyższymi standardami bezpieczeństwa, m.in. Certyfikat PCI DSS Level 1).

W przypadku korzystania z niniejszej funkcjonalności zalecane jest podpisanie żądania realizowanego do systemu Dotpay, jak i wymuszenie jego weryfikacji na danym sklepie ( id ). Opis realizacji podpisu zawarty został w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*.

W celu zarejestrowania danych karty klienta w systemie Dotpay należy w zleceniu płatności przesłać dodatkowe parametry:

#### 4.2.1 Tabela 4. (Parametry związane z rejestracją karty płatniczej)

PARAMETR	ZNACZENIE / OPIS
credit_card_store	<p>Parametr deklarujący opcję zapamiętania danych karty klienta po stronie Dotpay podczas dokonywania płatności.</p> <p>Dostępne wartości: 1 - wyrażenie zgody na zapisanie danych karty</p> <p>Przykład: credit_card_store = 1</p>
credit_card_customer_id	<p>Unikalny identyfikator płatącego nadany i przechowywany przez system sprzedawcy, wymagany podczas kolejnych płatności</p> <hr/> <p><b>Ważne:</b> Dotpay <b>NIE</b> zwraca tej wartości w żadnym miejscu (np. notyfikacje <i>URLC</i> czy <i>API</i>).</p> <hr/> <p>typ: <i>string</i> minimalna długość: 4 maksymalna długość: 1024</p> <p>Przykład: credit_card_customer_id = f9c6a4-25473</p>
credit_card_registration	<p>Parametr opcjonalny.</p> <p>Przesłanie spowoduje powstanie operacji typu credit_card_registration zamiast payment, co pozwala na zarejestrowanie karty w systemie bez dokonywania płatności.</p> <p>Zamiast obciążenia karty nastąpi chwilowa blokada środków na czas rejestracji w systemie, która zostanie automatycznie zwolniona po jej zakończeniu.</p> <hr/> <p><b>Wskazówka:</b> Dla operacji credit_card_registration wymagane jest użycie kwoty o wartości 1.00 ( amount = 1.00).</p> <hr/> <p>Dostępne wartości: true - rejestracja karty bez dokonywania płatności</p> <p>Przykład: credit_card_registration = true</p>

W celu realizacji płatności zarejestrowaną kartą należy w zleceniu płatności przesłać dodatkowe parametry opisane poniżej:

#### 4.2.2 Tabela 5. (Parametry wykorzystywane w procesie kolejnej płatności zarejestrowaną wcześniej kartą)

PARAMETR	ZNACZENIE / OPIS
credit_card_customer_id	<p>Unikalny identyfikator płatącego nadany przez system sprzedawcy opisany w powyższej tabeli.</p>

ciąg dalszy na następnej stronie



Table 5 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
credit_card_id	Identyfikator karty klienta zarejestrowanej po stronie systemu Dotpay Identyfikator może być przesłany w notyfikacji <i>URLC</i> (wymagana wcześniejsza konfiguracja konta, opis w rozdz. <i>Odbieranie informacji po płatności (powiadomienia urlc)</i> , alternatywnie może być pobrany ze szczegółów wykonanej operacji płatności za pośrednictwem <a href="#">API panelu administracyjnego sprzedawcy</a> <sup>18</sup>

Parametry wspólne dla realizacji zarówno pierwszej, jak i kolejnej płatności zostały przedstawione w tabeli poniżej, wszystkie są opcjonalne. Standardowo konfiguracja konta nie pozwala na ich wykorzystanie (stosowane są wtedy wartości domyślne, konfigurowalne po stronie Dotpay), możliwość ich przesłania może być zależna od specyficznych warunków umowy.

#### 4.2.3 Tabela 6. (Parametry wspólne dla pierwszej oraz kolejnej płatności)

<sup>18</sup> [https://www.dotpay.pl/developer/doc/api\\_seller/pl/#szczegoly-pojedynczej-operacji](https://www.dotpay.pl/developer/doc/api_seller/pl/#szczegoly-pojedynczej-operacji)

PARAMETR	ZNACZENIE / OPIS
credit_card_operation_type	<p>Parametr opcjonalny. Typ operacji. Dostępne wartości: e_commerce - pierwsza oraz kolejna płatność w modelu one-click (wartość domyślna), recurring_init - pierwsza transakcja, umożliwiająca późniejsze skorzystanie z płatności cyklicznych, recurring - płatność cykliczna (klient nie musi być obecny podczas obciążenia zapisanej wcześniej karty), Konto ( id ) może zostać skonfigurowane przez Dotpay, aby płatności cykliczne były domyślnym ustawieniem. W takim przypadku odpowiednie typy recurring_init oraz recurring będą nadawane automatycznie. Parametr jest wymagany w sytuacji, gdy jedno konto ( id ) obsługuje zarówno transakcje w modelu e-commerce jak i recurring.</p> <hr/> <p><b>Wskazówka:</b> Karta zarejestrowana poprzez e_commerce nie może zostać wykorzystana bezpośrednio do wykonania operacji typu recurring. Aby to było możliwe, wcześniej należy wykonać inicjalizację recurring czyli operację typu recurring_init. Jeśli karta została już zarejestrowana w procesie e_commerce, do wykonania recurring_init nie jest konieczne podawanie pełnych danych kartowych, wystarczy podać Identyfikator karty nadany przez system Dotpay ( credit_card_id ) Karta rejestrowana poprzez recurring_init może zostać wykorzystana zarówno do płatności typu recurring jak i e_commerce.</p> <hr/>
credit_card_security_code_required	<p>Parametr opcjonalny. Pozwala na sterowanie wymagalnością kodu zabezpieczającego CVV/CVV2 podczas płatności. Dotyczy jedynie kolejnej operacji e_commerce (one click). Dostępne wartości: yes - wymagaj (wartość domyślna), no - nie wymagaj, option - pole na kod pojawi się na formacie płatności, ale jego uzupełnienie jest nieobowiązkowe.</p>

ciąg dalszy na następnej stronie

Table 6 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
credit_card_threeds	<p>Parametr opcjonalny. Pozwala na sterowanie wymagalnością kodu autoryzacyjnego 3-D Secure. Dotyczy jedynie operacji e_commerce dla kart uczestniczących w programie.</p> <p>Dostępne wartości:                      yes - wymagaj (wartość domyślna, rekomendowana ze względów bezpieczeństwa transakcji),                      no - nie wymagaj,</p> <hr/> <p><b>Wskazówka:</b> Operacja typu recurring_init wymaga, natomiast recurring nie wymaga podania kodu autoryzacyjnego 3-D Secure, niezależnie od konfiguracji konta czy przesłanych parametrów.</p>
credit_card_avs	<p>Parametr opcjonalny. Pozwala na sterowanie wymagalnością dodatkowych danych osobowych, o które zostanie poproszony płacący na formacie płatności (dane adresowe oraz numer telefonu).</p> <p>Dostępne wartości:                      yes - wymagaj,                      no - nie wymagaj (wartość domyślna).</p>

#### 4.2.4 Przykładowe modele integracji oraz związane z nimi podstawowe wymagania)

##### Model integracji: 1 click

Table7: Tabela 7: Wymagania dla integracji typu 1 Click

Rejestracja karty	Kolejna płatność
Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 Proces płatności: - CVV - wymagane - 3DS - wymagane*	Wymagane parametry: - credit_card_customer_id - credit_card_id Proces płatności: - CVV - opcjonalne - 3DS - wymagane*

\* - zalecane ze względu na bezpieczeństwo transakcji

##### Model integracji: recurring

Table8: Tabela 8: Wymagania dla integracji typu *recurring*

Rejestracja karty	Kolejna płatność
Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 Proces płatności: - CVV - wymagane - 3DS - wymagane	Wymagane parametry: - credit_card_customer_id - credit_card_id Proces płatności: - CVV - nie dotyczy - 3DS - nie dotyczy

\* - zalecane ze względu na bezpieczeństwo transakcji

### Model integracji mieszany: 1 click + recurring

Table9: Tabela 9: Wymagania dla integracji typu *1 click + recurring* dla domyślnych ustawień konta: e-commerce

Model integracji	Rejestracja karty	Kolejna płatność
1 click	Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 Proces płatności: - CVV - wymagane - 3DS - wymagane*	Wymagane parametry: - credit_card_customer_id - credit_card_id Proces płatności: - CVV - opcjonalne - 3DS - wymagane*
recurring	Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 - credit_card_operation_type = recurring_init Proces płatności: - CVV - wymagane - 3DS - wymagane	Wymagane parametry: - credit_card_customer_id - credit_card_id - credit_card_operation_type = recurring Proces płatności: - CVV - nie dotyczy - 3DS - nie dotyczy

\* - zalecane ze względu na bezpieczeństwo transakcji

Table10: Tabela 10: Wymagania dla integracji typu *1 click + recurring* dla domyślnych ustawień konta: recurring

Model integracji	Rejestracja karty	Kolejna płatność
1 click	Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 - credit_card_operation_type = e_commerce Proces płatności: - CVV - wymagane - 3DS - wymagane*	Wymagane parametry: - credit_card_customer_id - credit_card_id - credit_card_operation_type = e_commerce Proces płatności: - CVV - opcjonalne - 3DS - wymagane*
recurring	Wymagane parametry: - credit_card_customer_id - credit_card_store = 1 Proces płatności: - CVV - wymagane - 3DS - wymagane	Wymagane parametry: - credit_card_customer_id - credit_card_id Proces płatności: - CVV - nie dotyczy - 3DS - nie dotyczy

\* - zalecane ze względu na bezpieczeństwo transakcji

Poniżej zostały zamieszczone przykładowe formularze przekierowania dla realizacji płat-

ności one-click.

Przykładowe żądanie płatności z możliwością rejestracji danych karty klienta:

```
1 <div>
2
3   <form action="https://ssl.dotpay.pl/t2/" method="post" id="dotpay_redirection_
  ↪form">
4     <input name="api_version" value="next" 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="11ac1938ac47ddd53815b4aeb6230ab9fe4554d82ee11e39c41b9055f38f5c08
  ↪" 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>
```

Przykładowe żądanie płatności z wykorzystaniem zarejestrowanych danych karty klienta (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="next" 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="ed0ef4e488ec2de3135b0a1ca226c31867f78bbcd8fe06506ae666210a78d68c
  ↪" type="hidden" />
```

(ciąg dalszy na następnej stronie)

```

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>

```

**Informacja:** W przypadku płatności cyklicznych zalecane jest stosowanie bezpośrednio komunikacji z Dotpay za pośrednictwem api restowego ( register order ).

Jeżeli dane karty płatniczej mają być podawane przez klienta po stronie sklepu sprzedawcy (np. w koszyku / podsumowaniu zamówienia zamiast na stronie płatności Dotpay), to w celu przekazania ich do systemu Dotpay należy wykorzystać parametry wymienione w poniższej tabeli.

**Informacja:** Przetwarzanie danych kart płatniczych po stronie systemu sprzedawcy wymaga, zgodnie z wytycznymi Payment Card Industry Data Security Standard (PCI DSS), dodatkowych certyfikacji.

W celu uzyskania szczegółowych informacji nt. wymaganych formalności należy skontaktować się z Działem Handlowym ([handlowy@dotpay.pl](mailto:handlowy@dotpay.pl)).

**Ostrzeżenie:** Przekazanie danych karty płatniczej za pomocą poniższych parametrów może być zrealizowane wyłącznie za pomocą metody *POST*.

#### 4.2.5 Tabela 11. (Parametry dla danych karty płatniczej)

PARAMETR	ZNACZENIE / OPIS
credit_card_number	Numer karty płatniczej klienta. typ: <i>credit_card_number</i> maksymalna długość: 26 wyrażenie regularne: <code>^\d\s{12,26}\$</code> Przykład: credit_card_number = 5500005555555559
credit_card_expiration_date_year	Rok daty ważności karty płatniczej klienta. długość: 4 wyrażenie regularne: <code>^(20)((19) ([2-9][0-9]))\$</code> Przykład: credit_card_expiration_date_year = 2019
credit_card_expiration_date_month	Miesiąc daty ważności karty płatniczej klienta. długość: 2 wyrażenie regularne: <code>^(0[1-9]) (1[0-2])\$</code> Przykład: credit_card_expiration_date_month = 02

ciąg dalszy na następnej stronie

Table 11 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
credit_card_security_code	Kod zabezpieczający karty płatniczej klienta (CVV2/CVC2). typ: <i>number</i> maksymalna długość: 4 wyrażenie regularne: <code>^\d{3,4}\$</code> Przykład: credit_card_security_code = 559

Wyrejestrowanie karty może nastąpić w następujące sposoby:

- 1) Klient może skorzystać z opcji wyrejestrowania, jaką Dotpay udostępnia w kierowanych do niego powiadomieniach mailowych o dokonaniu płatności.
- 2) Żądanie wyrejestrowania może zostać skierowane do Dotpay z systemu sprzedawcy.

W tym celu należy wykorzystać interfejs API udostępniony przez system Dotpay. Żądanie powinno zostać przesłane z wykorzystaniem metody *DELETE* na adres `https://ssl.dotpay.pl/t2/payment_api/v1/cards/{credit_card_id}/`, gdzie `{credit_card_id}` to identyfikator karty, która ma zostać wyrejestrowana.

Poniżej został przedstawiony przykład żądania (w języku PHP) i odpowiedzi wyrejestrowania karty. Dane użytkownika (user, password) autoryzujące żądanie są danymi wykorzystywanymi do logowania do panelu administracyjnego sprzedawcy.

Żądanie:

```

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($ch, CURLOPT_RETURNTRANSFER, 1);
10 curl_setopt($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>';

```

Przykładowe żądanie wyrejestrowania karty:

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

Odpowiedź:

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

Znaczenie zwracanych kodów odpowiedzi HTTP:

KOD	ZNACZENIE / OPIS
204 No Content <sup>19</sup>	Usunięto
404 Not Found <sup>20</sup>	Nie znaleziono karty
400 Bad Request <sup>21</sup>	Błąd podczas obsługi żądania

### 4.3 Płatność dzielona (Multimerchant / Pasaż)

**Informacja:** Funkcjonalność dostępna jest wyłącznie dla sklepów ( id ), które zostały odpowiednio skonfigurowane po stronie systemu Dotpay, co zależy jest od warunków wynikających z podpisanych przez akceptantów umów.

Funkcjonalność pozwala na podział wpłacanej przez klienta kwoty na dowolną ilość sklepów ( id ) w systemie Dotpay. Zlecenie podziału kwoty odbywa się poprzez przekazanie dodatkowych parametrów (obowiązkowe id(n) , amount(n) , oraz opcjonalne currency(n) , description(n) , control(n) , gdzie (n) jest liczbą całkowitą większą od zera, oznaczającą kolejne ID, kwoty itd.) przesyłanych wraz ze standardowymi parametrami opisanymi w rozdziale *wersja zaawansowana*.

**Ostrzeżenie:** Nie ma możliwości zrealizowania płatności Multimerchant dla wielu różnych walut jednocześnie, wartość wszystkich parametrów currency(n) musi być zgodna z currency .

W przypadku korzystania z niniejszej funkcjonalności podstawowy parametr id identyfikuje sklep, (tzw. multimerchant parent), na którym utworzona zostanie operacja typu payment\_multimerchant\_parent, informująca wyłącznie o pełnej kwocie płatności (przesłanej w parametrze amount ). Natomiast w sklepach id(n) (tzw. multimerchant child) zostaną zaksięgowane kwoty amount1 - amount(n) , widoczne jako operacje typu payment\_multimerchant\_child .

Dla każdego sklepu ( id ) dane przesyłane w notyfikacjach są stosowne do utworzonych na nich operacji payment\_multimerchant\_parent lub payment\_multimerchant\_child.

**Ostrzeżenie:** W przypadku kont multimerchant child adres URLC musi być wprowadzony w konfiguracji sklepu (*Ustawienia -> Powiadomienia -> Konfiguracja urlc -> Edycja*), tzn. nie ma możliwości zdefiniowania parametru typu urlc(n) .

**Ostrzeżenie:** Realizacja zwrotu środków możliwa jest wyłącznie dla pojedynczych transakcji payment\_multimerchant\_child zaksięgowanych na poszczególnych sklepach id(n) . **NIE** ma możliwości wykonania całościowego zwrotu poprzez operację payment\_multimerchant\_parent.

Poniżej został zamieszczony przykładowy formularz zlecenia płatności z podziałem kwoty.

1 <div>

2

(ciąg dalszy na następnej stronie)

<sup>19</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.5>

<sup>20</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

<sup>21</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1>



(kontynuacja poprzedniej strony)

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

---

**Ważne:** Suma kwot przesłanych w parametrach amount1, amount2, ..., amount(n) musi być równa całkowitej kwocie wpłacanej przez klienta, tj. kwocie przesłanej w parametrze amount .

---

## 4.4 Masscollect

---

**Informacja:** Funkcjonalność dostępna jest wyłącznie dla sklepów ( id ), które zostały odpowiednio skonfigurowane po stronie systemu Dotpay, co zależne jest od warunków wynikających z podpisanych przez akceptantów umów.

---

Funkcjonalność pozwala na przekazanie w zleceniu płatności numeru rachunku bankowego, na który ma zostać zrealizowana wypłata., tzn. domyślny numer wprowadzony w konfiguracji danego sklepu ( id ) zostanie nadpisany.

W zależności od konfiguracji konta, wypłaty zlecane przez włączony na koncie automat

zostaną zagregowane według przekazanych rachunków bankowych, bądź przekazane w schemacie 1:1, gdzie każda transakcja utworzy osobną operację wypłaty. Dla wypłat dowolnej kwoty wypłata zawsze zostanie zagregowana.

**Ostrzeżenie:** W przypadku korzystania z niniejszej funkcjonalności wymagane jest podpisanie żądania realizowanego do systemu Dotpay, jak i wymuszenie jego weryfikacji na danym sklepie ( id ). Opis realizacji podpisu zawarty został w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*.

W celu realizacji płatności typu Masscollect należy w zleceniu przesłać dodatkowe parametry opisane w tabeli poniżej. Wymagany jest jedynie recipient\_account\_number , przy czym zaleca się podanie również nazwy odbiorcy za pomocą recipient\_company lub recipient\_first\_name i recipient\_last\_name .

#### 4.4.1 Tabela 12. (Parametry dodatków wykorzystywane w usłudze Masscollect)

PARAMETR	ZNACZENIE / OPIS
recipient_account_number	Numer rachunku odbiorcy w formacie BBAN <sup>22</sup> . typ: <i>string</i> maksymalna długość: 26 Przykład: recipient_account_number = 32249000896640389235035459
recipient_company	Nazwa firmy odbiorcy płatności typ: <i>string</i> maksymalna długość: 50 Przykład: recipient_company = Moja Firma S.A.
recipient_first_name	Imię odbiorcy płatności typ: <i>string</i> maksymalna długość: 30 Przykład: recipient_first_name = Jan
recipient_last_name	Nazwisko odbiorcy płatności typ: <i>string</i> maksymalna długość: 30 Przykład: recipient_last_name = Kowalski
recipient_address_street	Ulica - adres odbiorcy płatności typ: <i>string</i> maksymalna długość: 40 Przykład: recipient_address_street = Wielicka
recipient_address_building	Numer budynku - adres odbiorcy płatności typ: <i>string</i> maksymalna długość: 10 Przykład: recipient_address_building = 72
recipient_address_apartment	Numer lokalu - adres odbiorcy płatności typ: <i>string</i> maksymalna długość: 10 Przykład: recipient_address_apartment = 1

ciąg dalszy na następnej stronie

Table 13 - kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
recipient_address_postcode	Kod pocztowy - adres odbiorcy płatności typ: <i>string</i> maksymalna długość: 6 Przykład: recipient_address_postcode = 30-552
recipient_address_city	Miasto - adres odbiorcy płatności typ: <i>string</i> maksymalna długość: 50 Przykład: recipient_address_city = Kraków

Poniżej został zamieszczony przykładowy formularz zlecenia płatności w modelu Masscollect.

```

1 <div>
2
3   <form action="https://ssl.dotpay.pl/t2/" id="dotpay_redirection_form" method=
  ↪ "POST" enctype="application/x-www-form-urlencoded">
4
5     <input type="hidden" name="id" value="123456">
6     <input type="hidden" name="amount" value="123.45">
7     <input type="hidden" name="currency" value="PLN">
8     <input type="hidden" name="api_version" value="next">
9     <input type="hidden" name="description" value="Płatność za zamówienie 07/2017">
10    <input type="hidden" name="recipient_account_number" value=
  ↪ "32249000896640389235035459">
11    <input type="hidden" name="recipient_company" value="Moja Firma S.A.">
12    <input type="hidden" name="recipient_first_name" value="Jan">
13    <input type="hidden" name="recipient_last_name" value="Kowalski">
14    <input type="hidden" name="recipient_address_street" value="Wielicka">
15    <input type="hidden" name="recipient_address_building" value="72">
16    <input type="hidden" name="recipient_address_apartment" value="1">
17    <input type="hidden" name="recipient_address_postcode" value="30-552">
18    <input type="hidden" name="recipient_address_city" value="Kraków">
19    <input type="hidden" name="chk" value="3135b6debc8fe4e488ec2easux506c31867f78bed0ef0a1ca2266
  ↪ " />
20  </form>
21
22  <p>
23    <button type="submit" form="dotpay_redirection_form" value="Submit">
24      Potwierdź zamówienie i zapłać
25    </button>
26  </p>
27
28 </div>

```

<sup>22</sup> [https://pl.wikipedia.org/wiki/Mi%C4%99dzynarodowy\\_numer\\_rachunku\\_bankowego](https://pl.wikipedia.org/wiki/Mi%C4%99dzynarodowy_numer_rachunku_bankowego)

## 4.5 Obsługa danych dostawy oraz płatącego

W celu poprawnej oceny zdolności kredytowej płatącego przez dostawcę danej metody płatności, do systemu Dotpay dodano dodatkowe parametry, które powinny zostać wysłane z systemu sprzedawcy przy inicjacji takiej płatności.

Poniżej prezentowany jest zestaw obsługiwanych parametrów wraz z opisem oraz ich wymagalnością.

Przesłanie większej liczby danych niż tylko „wymagane” dla danego kanału, może mieć duże znaczenie przy ostatecznej decyzji udzielenia kredytu. Dlatego, o ile to możliwe, zalecamy wysłanie kompletu danych.

W celu realizacji transakcji daną metodą płatności do integracji z Dotpay dodano nowy parametr `customer`. Parametr ten powinien zawierać konkretne informacje zgodnie z poniższą [Tabelą 13](#). Niektóre kanały płatności mogą wymagać mniejszej ilości danych, jak np. *PayPo* lub *Raty Alior*.

**4.5.1 Tabela 13. (Dane obsługiwane przez parametr `customer`)**

NAZWA POLA	TYP	OPIS
<code>payer.first_name</code>	string	Imię płatącego
<code>payer.last_name</code>	string	Nazwisko płatącego
<code>payer.email</code>	string	Adres email płatącego
<code>payer.phone</code>	string	Numer telefonu płatącego
<code>payer.address</code>	-	Adres płatącego / rozliczeniowy
<code>payer.address.city</code>	string	Adres płatącego: miasto
<code>payer.address.street</code>	string	Adres płatącego: ulica
<code>payer.address.building_number</code>	string	Adres płatącego: numer budynku
<code>payer.address.flat_number</code>	string	Adres płatącego: numer mieszkania
<code>payer.address.postcode</code>	string	Adres płatącego: kod pocztowy
<code>payer.address.country</code>	string	Adres płatącego: dwuliterowy (ISO 3166-1 alpha2 <sup>23</sup> ) lub trzyliterowy (ISO 3166-1 alpha3 <sup>24</sup> ) kod kraju
<code>is_logged_in</code>	boolean	Informacja czy płatący przed dokonaniem płatności był zarejestrowany w systemie sprzedawcy
<code>registered_since</code>	string	Data rejestracji płatącego w serwisie sprzedawcy, format YYYY-MM-DD lub YYYY-MM-DD hh:mm:ss Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: <code>order_count</code>
<code>registered_since_indicator</code>	string ( <i>indykator</i> )	Data rejestracji płatącego w serwisie sprzedawcy, indyikator dla pola <code>registered_since</code> Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: <code>order_count</code>
<code>account_update</code>	string	Data ostatniej zmiany konta płatącego w serwisie sprzedawcy, format YYYY-MM-DD
<code>account_update_indicator</code>	string ( <i>indykator</i> )	Data ostatniej zmiany konta płatącego w serwisie sprzedawcy, indyikator dla pola <code>account_update</code>
<code>password_change</code>	string	Data ostatniej zmiany hasła dla konta płatącego w serwisie sprzedawcy, format YYYY-MM-DD
<code>password_change_indicator</code>	string ( <i>indykator</i> )	Data ostatniej zmiany hasła dla konta płatącego w serwisie sprzedawcy, indyikator dla pola <code>password_change</code>
<code>shipping_address_since</code>	string	Data od kiedy podany adres dostawy płatącego w serwisie sprzedawcy jest używany, format YYYY-MM-DD

ciąg dalszy na następnej stronie

Table 14 - kontynuacja poprzedniej strony

NAZWA POLA	TYP	OPIS
shipping_address_since_indicator	string (indicator)	Data od kiedy podany adres dostawy płatącego w serwisie sprzedawcy jest używany, indyikator dla pola shipping_address_since
order_count	int	Ilość złożonych zamówień przez płatącego w serwisie sprzedawcy od daty rejestracji Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr registered_since
order_count_day	int	Ilość złożonych zamówień przez płatącego w serwisie sprzedawcy w tym samym dniu
order_count_year	int	Ilość złożonych zamówień przez płatącego w serwisie sprzedawcy w tym samym roku
fraud_activity	boolean	Czy sklep kiedykolwiek zanotował podejrzaną aktywność na koncie tego kupującego
order_history	-	Dotychczasowa historia zakupowa płatącego w sklepie
order_history.date	string	Data zakupu n-tego produktu, format YYYY-MM-DD
order_history.amount	decimal (10,2)	Cena brutto n-tego produktu w sklepie internetowym
order	-	Zamówienie
order.total_amount	string	Wartość całego zamówienia
order.id	string	Identyfikator zamówienia w systemie sprzedawcy. Odpowiada numerowi ID całego zamówienia w bazie sklepu
order.items	-	Zawartość koszyka zakupowego Zalecamy nie przekraczać liczby 100 artykułów w przesłanej liście
order.items.id	string	Identyfikator pojedynczego produktu w systemie sprzedawcy. Odpowiada numerowi ID produktu w bazie sklepu
order.items.name	string	Nazwa pojedynczego produktu w systemie sprzedawcy. Odpowiada nazwie produktu w bazie danych sklepu. minimalna długość: 1 maksymalna długość: 150 wyrażenie regularne: <code>[\\w\\s\\-\\. ,'?@\\ \\/ąćęłńóśźżĄĆĘŁŃÓŚŹŻ]+\$</code>
order.items.quantity	int	Ilość pozycji w zamówieniu klienta, czyli: n * Towar = łączna ilość pozycji minimalna ilość: 1 maksymalna ilość: 1000
order.items.unit_type	string	Jednostka miary ilości sztuk n-tego produktu w systemie sprzedawcy. np. sztuki, kg, cm, litry, itp...
order.items.gross_price	decimal(10,2)	Cena brutto n-tego produktu w systemie sprzedawcy. Odpowiada cenie produktu w bazie danych sklepu.
order.items.type	string	Typ zamówienia (np. towar, płatność, dostawa, rabat)
order.items.is_virtual	boolean	Produkt wirtualny (nie wymaga dostawy)
order.items.category	string	Kategoria pojedynczego produktu w systemie sprzedawcy. kategoria powinna być zgodna z jednym z wymienionych w słowniku elementem. Spis dostępnych kategorii: <i>Słownik dostępnych kategorii sprzedażowych</i>

ciąg dalszy na następnej stronie

Table 14 - kontynuacja poprzedniej strony

NAZWA POLA	TYP	OPIS
order.delivery_type	string	Metoda dostawy Dostępne wartości: <ul style="list-style-type: none"> <li>• COURIER - kurier</li> <li>• POCZTA_POLSKA - Poczta Polska</li> <li>• PICKUP_POINT - dostawa do punktu (np. UPS Access point, DHL Parcel Shop)</li> <li>• PACZKOMAT - paczkomat</li> <li>• PACZKA_W_RUCHU - paczka w ruchu</li> <li>• PICKUP_SHOP - odbiór w sklepie (click&amp;collect)</li> </ul>
order.delivery_address	-	Adres dostawy <b>Jeśli paczka jest dostarczana do punktu/paczkomatu/itd, to taki adres i nazwa powinna być podana a nie dane faktycznego odbiorcy.</b>
order.delivery_address.city	string	Adres dostawy: miasto
order.delivery_address.street	string	Adres dostawy: ulica
order.delivery_address.building_number	string	Adres dostawy: numer budynku
order.delivery_address.flat_number	string	Adres dostawy: numer mieszkania
order.delivery_address.postcode	string	Adres dostawy: kod pocztowy
order.delivery_address.country	string	Adres dostawy: dwuliterowy (ISO 3166-1 alpha2 <sup>25</sup> ) lub trzyliterowy (ISO 3166-1 alpha3 <sup>26</sup> ) kod kraju
order.delivery_address.name	string	Nazwa odbiorcy/punktu odbiorczego. Przykłady: order.delivery_address.name = D0B019A order.delivery_address.name = PPP:6252652
order.delivery_address.phone	string	Numer telefonu odbiorcy
order.delivery_address.is_verified	bool	Adres dostawy: czy adres dostawy jest zweryfikowany

**Informacja:** W przypadku gdy sklep nie chce przekazywać właściwej daty, dla wybranych parametrów możliwe jest skorzystanie z pola zastępczego typu indykator.

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

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

<sup>25</sup> [https://pl.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-2](https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-2)

<sup>26</sup> [https://pl.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-3](https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-3)

#### 4.5.2 Wartości używane dla pola zastępczego typu indyktor dla wybranych pól:

WARTOŚĆ	OPIS
01	Konto płacącego w serwisie sprzedawcy nie istnieje
02	Data zlecanej właśnie transakcji
03	Data nie starsza niż 30 dni temu
04	Data w przedziale 30 - 60 dni temu
05	Data starsza niż 60 dni temu

Poniżej przykładowe dane z użyciem powyższych parametrów:

Listing 9: Przykład użycia parametrów (format json)

```
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.",
```

(ciąg dalszy na następnej stronie)

```

48         "gross_price": "15.00",
49         "type": "dostawa",
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 Słownik kategorii artykułów dla parametru order.items.category

WARTOŚĆ	OPIS
ART_ANTIQUES	Dzieła sztuki, antyki
BOOKS_EDUCATIONAL	Książki, materiały edukacyjne
BUILDING_MATERIALS	Materiały budowlane
CARS_MOTORBIKES_SCOOTERS	Samochody, motory, motorowery, skutery
CLOTHING_SHOES	Odzież, buty
COMP_COMPONENTS	Podzespoły komputerowe i akcesoria w tym monitory, drukarki i skanery
COMPUTERS	Komputery
CONSOLES_GAMES	Konsole i gry
DEVOTIONAL	Dewocjonaalia
DOORS_WINDOWS	Okna i drzwi
ECO_SYSTEMS	Systemy ekologiczne
FOR_CHILDREN	Wózki dziecięce, foteliki samochodowe i inne
FUEL_COAL	Opał i węgiel
FURNITURE	Meble
GARDENING	Nasiona, nawozy, sadzonki
GIFT_VOUCHERS	Bony upominkowe, pakiety (inne niż medyczne)
HEATING	Kotły CO, JUNKERSY
HOME_FURNISHINGS	Wyposażenie domu
HOME_TEXTILES	Tekstylia domowe, pościel i koce
HOUSEHOLD_GOODS_BIG	AGD duże
HOUSEHOLD_GOODS_SMALL	AGD małe
INSURANCE	Usługi ubezpieczeniowe
JEWELRY_WATCHES	Biżuteria, zegarki
LAPTOPS_TABLETS	Laptopy i tablety
MUSICAL_INSTRUMENTS	Instrumenty muzyczne
OTHER_MOTORIZATION	Pozostała motoryzacja
PHONES_GPS	Telefony i GPS
PHOTO_CAMERA	Foto i kamery
RTV	RTV
SERVICES	Usługi
SOFTWARE	Oprogramowanie i inne komponenty komputerowe
SPORTS_REHABILITATION	Artykuły sportowe/rehabilitacyjne/turystyczne

ciąg dalszy na następnej stronie



Table 16 – kontynuacja poprzedniej strony

WARTOŚĆ	OPIS
TOOLS_DEVICES	Narzędzia i urządzenia
VEHICLE_ACCESSORIES_EQUIPMENT	Akcesoria i wyposażenie pojazdu

#### 4.5.4 Wymagane dane przesyłane w parametrze customer dla kanału PayPal

W celu realizacji płatności z wykorzystaniem kanału PayPal, należy wysłać komplet danych wymaganych przez ten kanał płatności. Wybrane dane z *Tabeli 13* niezbędne do realizacji płatności tym kanałem:

Listing 10: Przykład użycia minimalnej ilości wymaganych danych dla kanału PayPal (format 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 11: Przykład użycia kompletnych danych dla kanału **PayPo** (format 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 }

```

Kolorem oznaczono parametry opcjonalne dla kanału **PayPo**. Nie są one wymagane, ale ich brak może negatywnie wpłynąć na ocenę zdolności kredytowej kupującego.

Aby takie dane mogły być przesłane do Dotpay w parametrze customer, należy je

odpowiednio wcześniej sformatować. Dane te powinny zostać zakodowane do formatu JSON<sup>27</sup> a następnie zakodowane przy użyciu Base64<sup>28</sup>.

Listing 12: Przykład przygotowania parametru customer dla kanału **PayPo** (wersja minimum) w języku 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 Wymagane dane przesyłane w parametrze customer dla kanału Raty Alior

W celu realizacji płatności z wykorzystaniem kanału Raty Alior, należy wysłać komplet danych wymaganych przez ten kanał płatności. Wybrane dane z *Tabeli 13* niezbędne do realizacji płatności tym kanałem:

Listing 13: Przykład użycia minimalnej ilości wymaganych danych dla kanału Raty Alior (format json):

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

(ciąg dalszy na następnej stronie)

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

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

(kontynuacja poprzedniej strony)

```
9           {
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     }
```

Wartość parametru `order.items.category` powinna być wypełniona zgodnie ze *słownikiem kategorii*.

Ilość produktów w parametrze `order.items` jest ograniczona i nie powinna być większa niż 500 elementów.

Aby takie dane mogły być przesłane do Dotpay w parametrze `customer`, należy je odpowiednio wcześniej sformatować. Dane te powinny zostać zakodowane do formatu JSON<sup>29</sup> a następnie zakodowane przy użyciu Base64<sup>30</sup>.

Listing 14: Przykład przygotowania parametru `customer` dla kanału **Raty Alior** (wersja minimum) w języku 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 Phone 1",
14                "quantity" => 1,
15                "gross_price" => "1200.00",
16                "category" => "PHONES_GPS"
17            ),
18            array(
19                "name" => "Pendrive 64GB",
20                "quantity" => 4,
21                "gross_price" => "50.00",
22                "category" => "COMP_COMPONENTS"
23            )
24        ]
25    ),
26 ),
27
28
29
```

(ciąg dalszy na następnej stronie)

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

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

```

30     );
31
32     $customer_base64 = base64_encode(json_encode($customer));
33
34     ?>
35

```

#### 4.5.6 Wymagane dane przesyłane w parametrze customer dla kanału PayPal (tylko z programem SPP w PayPal)

Wymagania techniczne niezbędne do uzyskania „Rozszerzonej polityki ochrony sprzedających (SPP – Seller Protection Policy)” od PayPal dla Kup online – odbiór w sklepie lub punkcie.

Jeśli korzystasz z programu ochrony sprzedających na swoim koncie PayPal i posiadasz stosowną umowę w tym zakresie z PayPal, w integracji z Dotpay konieczne jest przesłanie dodatkowych informacji.

W celu realizacji płatności z wykorzystaniem kanału PayPal w programie SPP należy wysłać komplet danych wymaganych przez ten kanał płatności. Wybrane dane z *Tabeli 13* niezbędne do realizacji płatności tym kanałem:

Listing 15: Przykład użycia minimalnej ilości wymaganych danych dla kanału PayPal w programie SPP (format json):

```

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    }

```

Aby takie dane mogły być przesłane do Dotpay w parametrze customer, należy je odpowiednio wcześniej sformatować. Dane te powinny zostać zakodowane do formatu JSON<sup>31</sup> a następnie zakodowane przy użyciu Base64<sup>32</sup>.

Listing 16: Przykład przygotowania parametru customer dla kanału **PayPal** (SPP) w języku PHP:

```

1     <?php
2
3     $customer = array(
4         "payer" => array(

```

(ciąg dalszy na następnej stronie)

<sup>31</sup> <https://en.wikipedia.org/wiki/JSON>

<sup>32</sup> <https://pl.wikipedia.org/wiki/Base64>

```

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

```

---

**Informacja:** Aby przesłane dane były prawidłowo interpretowane i wysłane z Dotpay do PayPal, należy wcześniej zgłosić taki fakt do Dotpay ([tech@dotpay.pl](mailto:tech@dotpay.pl)) w celu prawidłowej konfiguracji konta.

---

## 4.6 Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy

Po stronie systemu Dotpay istnieje możliwość konfiguracji sklepu ( id ) w taki sposób, aby w przypadku zaistnienia błędu przejścia do Dotpay (spowodowanego np. nieprawidłowym przekazaniem parametrów przez system sprzedawcy) nastąpiło automatyczne przekierowanie na adres z parametru url (przesłany przez system sprzedawcy) wraz z przekazaniem kodu błędu.

W sytuacji zaistnienia błędu nastąpi przekierowanie na adres z parametru url , do którego zostanie dołączony parametr error\_code z odpowiednią wartością.

---

**Informacja:** W celu aktywacji funkcjonalności należy zaznaczyć opcję *Handling error codes in URL*, dostępną w panelu administracyjnym sprzedawcy, w zakładce *Ustawienia -> Konfiguracja sklepu -> Edycja* .

---

Przykładowo, w przypadku przekierowania z systemu sprzedawcy na nieznaną numer kanału płatności w Dotpay, dla przekazanego również w przekierowaniu parametru url : url = <https://example.com/>

nastąpi przekierowanie na adres: [https://example.com/?error\\_code=UNKNOWN\\_CHANNEL](https://example.com/?error_code=UNKNOWN_CHANNEL)

Znaczenie wartości parametru error\_code zostało opisane poniżej:

PAYMENT\_EXPIRED - przekroczona data ważności linku płatniczego lub przekazana w parametrze expiration\_date

UNKNOWN\_CHANNEL - nieprawidłowa wartość parametru channel

DISABLED\_CHANNEL - wskazany kanał płatności jest niedostępny

UNKNOWN\_CURRENCY - kod waluty jest nieprawidłowy

BLOCKED\_ACCOUNT - konto ( id ) jest zablokowane

INACTIVE\_SELLER - konto ( id ) nie jest aktywne

AMOUNT\_TOO\_LOW - kwota jest mniejsza niż minimum określone dla sklepu

AMOUNT\_TOO\_HIGH - kwota jest większa niż maksimum określone dla sklepu

BAD\_DATA\_FORMAT - przesłane dane posiadają nieprawidłowy format np. błędny format parametru `expiration_date`

URLC\_INVALID - ustawienia konta ( id ) w Dotpay wymagają by adres URLC zawierał protokół SSL (adres rozpoczyna się od: «`https://`»)

REQUIRED\_PARAMETERS\_NOT\_PRESENT - brak jednego z wymaganych parametrów

MULTIMERCHANT\_INVALID\_ACCOUNT\_CONFIGURATION - jedno z kont nie jest odpowiednio skonfigurowane po płatności typu Multimerchant

MULTIMERCHANT\_INSUFFICIENT\_AMOUNT - suma wartości parametrów `amount(n)` jest niezgodna z wartością `amount`

MULTIMERCHANT\_WRONG\_CURRENCY - wartości parametrów `currency` oraz `currency(n)` są niezgodne

CREDIT\_CARD\_NOT\_ACCEPTED - przesłano dane kartowe, ale konfiguracja konta ( id ) nie pozwala na ich przetwarzanie

CREDIT\_CARD\_OPERATION\_TYPE\_NOT\_ACCEPTED\_CODE - przesłano parametr `credit_card_operation_type` , ale konfiguracja konta ( id ) nie pozwala na jego przetwarzanie

CREDIT\_CARD\_SECURITY\_CODE\_REQUIRED\_NOT\_ACCEPTED\_CODE - przesłano parametr `credit_card_security_code_required` , ale konfiguracja konta ( id ) nie pozwala na jego przetwarzanie

CREDIT\_CARD\_3DS\_NOT\_ACCEPTED\_CODE - przesłano parametr `credit_card_threeds` , ale konfiguracja konta ( id ) nie pozwala na jego przetwarzanie

CREDIT\_CARD\_AVS\_NOT\_ACCEPTED\_CODE - przesłano parametr `credit_card_avs` , ale konfiguracja konta ( id ) nie pozwala na jego przetwarzanie

UNKNOWN\_ERROR - wartość zwracana w innym przypadku niż powyższe

## 5 ŚRODOWISKO TESTOWE

W serwisie Dotpay istnieje możliwość utworzenia środowiska testowego (w pełni niezależnego od środowiska produkcyjnego), które pozwala na wykonywanie symulacji operacji (transakcji) celem przeprowadzenia testów sklepu integrowanego z systemem płatności.

---

### Informacja:

W celu uzyskania dostępu do środowiska testowego należy wypełnić formularz rejestracji znajdujący się pod adresem

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

---

Adresy środowiska testowego zostały wymienione poniżej, natomiast wszelkie działania, jakie należy wykonać w celu dokonania integracji z systemem Dotpay są analogiczne do opisanych w powyższych rozdziałach.

[https://ssl.dotpay.pl/test\\_payment/](https://ssl.dotpay.pl/test_payment/) - adres formularza płatności

[https://ssl.dotpay.pl/test\\_seller/](https://ssl.dotpay.pl/test_seller/) - adres logowania do panelu administracyjnego

[https://ssl.dotpay.pl/test\\_payment/payment\\_api/channels/](https://ssl.dotpay.pl/test_payment/payment_api/channels/) - adres API płatności (lista kanałów)

Podczas symulacji płatności kartą (kanał 248) wymagane będzie wprowadzenie odpowiednich danych.

Poniżej znajduje się lista przykładowych kart, które można wykorzystać do tego celu. Data ważności musi być przyszła, ale nie późniejsza niż grudzień 2030.

## 5.1 Tabela 14. (Lista przykładowych numerów kart, które można wykorzystać w procesie testowania płatności kartowych)

TYP	NUMER	CVV2 / CVC2	3DS
Visa*	4916 9715 6289 1025	025	Nie
Visa*	4929 5320 2788 7670	670	Tak
MasterCard*	5498 5400 7907 4343	343	Nie
MasterCard*	5344 6642 8071 1026	026	Tak
Visa	4111 1111 1111 1111	111	Nie
Visa	4444 4444 4444 4448	448	Tak
MasterCard	5500 0055 5555 5559	559	Nie

\* przeznaczone do testów usług one-click oraz recurring

## 6 INFORMACJE DODATKOWE

### 6.1 Weryfikacja adresu IP

Adresy IP serwisu Dotpay:

195.150.9.37 oraz 91.216.191.181 - 91.216.191.185 i 5.252.202.255

Z powyżej wymienionych adresów system Dotpay przesyła powiadomienia do serwisów sprzedawców. Aby komunikacja była możliwa, system sprzedawcy powinien akceptować połączenia z niniejszych adresów.

**Ostrzeżenie:** W celu zapewnienia własnego bezpieczeństwa system sprzedawcy powinien zawsze weryfikować adres IP serwisu Dotpay, z jakiego zostało przesłane powiadomienie.

Ze względu na wymogi PCI DSS jedyną obsługiwaną przez serwery Dotpay wersją SSL jest *TLSv1.2* oraz *TLSv1.3*, w przypadku pozostałych protokołów szyfrowane połączenie z hostem [ssl.dotpay.pl](https://ssl.dotpay.pl) nie będzie możliwe.

## 6.2 Bezpieczeństwo integracji płatności

W celu zapewnienia najwyższych standardów bezpieczeństwa transakcji wykonywanych za pośrednictwem Dotpay zalecamy stosowanie wszystkich dostępnych mechanizmów zabezpieczających proces płatności.

---

**Informacja: O bezpieczeństwie transakcji należy pamiętać na dwóch etapach:**

- *Przekierowanie kupującego z serwisu Sprzedawcy do Dotpay*
  - *Odbioru potwierdzenia o zaksięgowanej wpłacie w serwisie Dotpay*
- 

### 6.2.1 Przekierowanie kupującego do Dotpay

Przekierowanie kupującego do serwisu Dotpay wiąże się z przesłaniem odpowiednich parametrów określających szczegóły transakcji. Aby wyeliminować nieuprawnioną manipulację danymi przed przekazaniem ich do Dotpay, zalecamy ochronę integralności parametrów przekierowania realizowanego z serwisu Sprzedawcy. System Dotpay umożliwia podpisywanie wartości przesyłanych parametrów. Podpis powinien zostać przekazany, jako dodatkowy parametr `chk`, (wraz z resztą parametrów żądania kierowanego do strony płatności systemu Dotpay).

Opis tej funkcjonalności znajduje się w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*.

W przypadku gotowości ze strony Sprzedawcy na przesyłanie dla wszystkich transakcji parametru `chk`, prosimy o przekazanie do Dotpay takiej informacji w celu weryfikacji poprawności integracji płatności.

**Ostrzeżenie:** Należy pamiętać, że minimalna ilość parametrów przesłana do Dotpay inicjujących płatność to: `id`, `amount`, `currency`, `description`.

Prosimy o weryfikację czy integracja płatności pozwala na poprawne przesłanie tych parametrów do Dotpay w formacie zgodnym z niniejszą dokumentacją.

### 6.2.2 Odbiór potwierdzenia o zaksięgowanej wpłacie w serwisie Dotpay

W celu przekazywania do serwisu sprzedawcy informacji o dokonanej operacji (transakcji) istnieje *mechanizm powiadomień URLC* (HTTP request, połączenie asynchroniczne, callback), które wysyłane są za pomocą metody `POST` oraz w pełni niezależnie od działań kupującego.

Jeśli integracja płatności wykorzystuje automatyczne notyfikacje *URLC*, zwracamy uwagę by dla każdego zamówienia weryfikować przesłane do sprzedawcy w notyfikacji dane, a w szczególności typ operacji (`operation_type`), jej status (`operation_status`) oraz: - `operation_original_amount` - kwota transakcji pobrana z parametru `amount`, jaki został przesłany przez serwis sprzedawcy w przekierowaniu Kupującego do serwisu Dotpay - `operation_original_currency` - waluta operacji (transakcji) pobrana z parametru `currency` jaki został przesłany przez serwis sprzedawcy w przekierowaniu Kupującego do serwisu Dotpay - `signature` - suma kontrolna będąca wynikiem działania funkcji skrótu SHA-256 z przesłanych w notyfikacji *URLC* do Sprzedawcy parametrów

Zalecamy również by każda notyfikacja *URLC* odbierana z Dotpay była weryfikowana pod kątem poprawności źródła, zatem warto zweryfikować czy została przysłana z właściwego dla Dotpay **adresu IP** oraz czy została ona dostarczona metodą `POST`. Adres IP, z którego Dotpay wysłała notyfikację jest wymieniony w rozdziale *Weryfikacja adresu IP*.



**Ostrzeżenie:** Po odebraniu potwierdzenia zaksięgowania wpłaty w serwisie Dotpay, sklep bezwzględnie powinien porównać wartość zaksięgowanej kwoty i waluty z oczekiwaną kwotą zarejestrowaną przy składaniu zamówienia przez kupującego w bazie danych sklepu.

**Zamówienie powinno być realizowane przez sklep jedynie po potwierdzeniu zgodności wartości należnej za zamówienie zapisane w bazie sklepu z wartością zaksięgowaną w Dotpay przypisaną do danego zamówienia!**

## 7 ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI)

W poniższej tabeli znajduje się aktualna lista kanałów udostępnianych w systemie Dotpay. Dostępność danej metody płatności na danym koncie ( id ) zależy od podpisanej umowy.

W tabeli zostały przedstawione numery i nazwy kanałów, ich dostawcy, logotypy oraz uwagi. Numery kanałów są wartościami, jakie przyjmuje parametr channel , opisany we wcześniejszych rozdziałach.


### 7.1 Tabela 15. (Kanały płatności dostępne w serwisie Dotpay)

#### 7.1.1 KARTY PŁATNICZE









NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
71	MasterPass	-		24/7	Dostępny wyłącznie dla kont firmowych.
246	Karty płatnicze	-		24/7	Dostępny wyłącznie dla kont firmowych.
248	Karty płatnicze	-		24/7	Dostępny wyłącznie dla kont firmowych. Dostępny dla walut: PLN, EUR, USD, GBP
249	Visa SRC	-		24/7	Dostępny wyłącznie dla kont firmowych.

ciąg dalszy na następnej stronie

Table 18 - kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
260	Google Pay	Google LLC		24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa.
262	Apple Pay	Apple Inc.		24/7	Dostępny wyłącznie dla kont firmowych.

### 7.1.2 SZYBKIE TRANSFERY

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
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 (dawniej 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	Płacę z Alior Bankiem	Alior Bank S.A.		24/7	
46	Płacę z Citi Handlowy	Citi Bank Handlowy S.A.		24/7	

ciąg dalszy na następnej stronie

Table 19 - kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
50	Pay Way Toyota Bank	Toyota Bank Polska		24/7	
51	Płać z BOŚ	BOŚ Bank S.A.		24/7	
66	Bank Nowy S.A.	Bank Nowy Spółka Akcyjna z siedzibą w Poznaniu		24/7	
70	Pocztowy24	Bank Pocztowy S.A.		24/7	
73	BLIK	Polski Standard Płatności Sp. z o.o.		24/7	
74	Banki Spółdzielcze	Krajowa Izba Rozliczeniowa S.A.		24/7	
75	Płać z Plus Bank	Krajowa Izba Rozliczeniowa S.A.		24/7	
76	VeloBank PBL	Krajowa Izba Rozliczeniowa S.A.		24/7	
86	TrustPay	Trust Pay		24/7	Dostępny wyłącznie dla kont firmowych. Dostępny dla walut: CZK, EUR
87	Credit Agricole PBL	Credit Agricole Bank Polska S.A.		24/7	Dostępny wyłącznie dla kont firmowych.
90	BNP Paribas - płać z Pl@net	Krajowa Izba Rozliczeniowa S.A.		24/7	Dostępny wyłącznie dla kont firmowych.

ciąg dalszy na następnej stronie



Table 19 - kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
91	Nest Bank	Krajowa Izba Rozliczeniowa S.A.		24/7	Dostępny wyłącznie dla kont firmowych.
92	Bank Spółdzielczy w Brodnicy	Krajowa Izba Rozliczeniowa S.A.		24/7	Dostępny wyłącznie dla kont firmowych.
93	Kasa Stefczyka	Spółdzielcza Kasa Oszczędnościowo-Kredytowa im. F. Stefczyka		24/7	

### 7.1.3 PRZELEWY ONLINE

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
7	ING Klienci korporacyjni	ING Bank Śląski S.A.		pon. - sob. 8:00 - 20:00	
10	Millennium Klienci korporacyjni	Millennium Bank S.A.		pon. - pt. 8:00 - 20:00	
15	iPKO	Bank PKO BP	Powszechna Kasa Oszczędności Bank Polski S.A.	0:00 - 23:00 / 7	
16	Credit Agricole	Credit Agricole Bank Polska S.A.		4:00 - 23:00 / 7	
32	BNP Paribas	BNP Paribas Bank Polska SA		pon. - pt. 8:00 - 21:00	
89	Santander	Santander Bank Polska SA (dawniej Bank Zachodni WBK SA)		pon. - pt. 8:00 - 21:00	



#### 7.1.4 PŁATNOŚCI GOTÓWKOWE

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
11	Przelew/Przekaz	-	 Przekaz przelew bankowy	pon. - pt. 3 sesje elixir	
82	Przelew SEPA	-	 SEPA Single Euro Payments Area Einheitlicher Euro-Zahlungsverkehrsräum	pon. - pt. 3 sesje elixir	Dostępny wyłącznie dla kont firmowych. Dostępny dla walut: EUR.


#### 7.1.5 PORTMONETKI I VOUCHERY

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
52	SkyCash	SkyCash Poland S.A.	 SkyCash™	24/7	
59	CinkciarzPAY	Conotoxia Sp. z o.o.	 € \$ PAY	24/7	
218	paysafecard	Paysafecard	 paysafecard	24/7	Dostępny dla walut: PLN UWAGA: dla tego kanału brak możliwości zlecenia zwrotu



#### 7.1.6 RATY

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
55	Raty z Alior Bankiem	Alior Bank S.A.		24/7	Dostępny wyłącznie dla kont firmowych, dla płatności w przedziale od 300.00 PLN do kwoty ustalonej w Umowie.
68	mRaty	mBank S.A.		24/7	Dostępny wyłącznie dla kont firmowych, dla płatności w przedziale od 300.00 PLN do kwoty ustalonej w Umowie.



#### 7.1.7 INNE

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
212	PayPal	PayPal		24/7	Dostępny dla walut: PLN Model Gateway Dotpay nie przesyła środków do sprzedawcy.

#### 7.1.8 PŁATNOŚCI ODROCZONE



NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
94	Kupuj teraz zapłać później	Aiqlabs Sp. z o.o.		24/7	Dostępny dla płatności w przedziale od 100.00 PLN do 2000.00 PLN lub do kwoty ustalonej w Umowie.
95	PayPo	PayPo Sp. z o.o.		24/7	Dostępny wyłącznie dla kont firmowych dla płatności w przedziale od 10.00 PLN do 2000.00 PLN. Wymagana oddzielna umowa. Niezbędna w integracji tego kanału obsługa parametru customer .

### 7.1.9 PŁATNOŚCI MOBILNE typu Direct Carrier Billing

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
231	Orange	Orange Polska S.A.		24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.
232	T-Mobile	T-Mobile Polska S.A.		24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.

ciąg dalszy na następnej stronie

Table 26 - kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
233	PLAY	P4 Sp. z o.o.		24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.
234	Plus	Polkomtel Sp. z o.o.		24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.

## 7.2 Kanały płatności - do pobrania

**Informacja:** Lista wymienionych kanałów oraz ich logotypy, dostępne są do pobrania:

- Logotypy kanałów płatności: 1.94.1.0: [format zip](#)
- Lista dostępnych kanałów płatności: 1.94.1.0: [format xlsx](#)
- Lista dostępnych kanałów płatności: 1.94.1.0: [format json](#)

## 8 ZAŁĄCZNIK II (OPISY STATUSÓW OPERACJI)

W poniższej tabeli zostały przedstawione opisy statusów operacji tworzonych w systemie.

### 8.1 Tabela 16. (Statusy operacji tworzone w systemie Dotpay)

STATUS	ZNACZENIE / OPIS
new (nowa)	Nowa operacja. Oznacza tylko i wyłącznie fakt wystąpienia operacji danego typu. W tym stanie nie zawiera żadnych księgowania.
processing (oczekuje na wpłatę)	Operacja przetwarzana, np. dla typu payment oznacza, że płacący wrócił na stronę Dotpay od dostawcy kanału płatności, bądź dostawca kanału płatności poinformował Dotpay o tym fakcie. Operacje typu payment i payment_multimerchant_child nie posiadają księgowania w tym stanie. Operacja typu complaint w tym stanie oznacza rozpoczęcie procedury reklamacji.

ciąg dalszy na następnej stronie



Table 27 – kontynuacja poprzedniej strony

STATUS	ZNACZENIE / OPIS
completed (wykonana)	<p>Dla typów payment, payment_multimerchant_child oznacza, że Dotpay już posiada (bądź ma pewność, że będzie posiadał) środki na swoich kontach z tytułu dokonanej przez klienta płatności.</p> <p>Dla typu release_rollback oznacza, że zablokowane środki z tytułu rollback wróciły na konto ( id ) sklepu.</p> <p>Dla typów payout, refund i complaint oznacza, że środki zostały przekazane na konto akceptanta, do płacącego czy dostawcy kanału płatności. Jest to stan ostateczny, operacja już nie zmieni swojego statusu.</p> <p>Do operacji mogą pojawić się dodatkowe księgowania, ale tylko takie, które są skutkiem np. korekty, źle naliczonej korekty.</p>
rejected (odrzucona)	<p>Dla typów payment i payment_multimerchant_child oznacza, że Dotpay <b>NIE</b> dostał środków z tytułu płatności i ma pewność, że ich nie dostanie. Oznacza to, że np. płacący zrezygnował z płatności, bądź nie posiada środków na koncie u dostawcy kanału płatności na realizację tej operacji.</p> <p>Dla typów payout, refund i complaint oznacza, że operacja została anulowana – środki wróciły na konto ( id ) sklepu. Jest to stan ostateczny, operacja już nie zmieni statusu.</p> <p>Do operacji mogą pojawić się dodatkowe księgowania, ale tylko takie, które są skutkiem np. korekty, źle naliczonej korekty.</p>
process- ing_realization_waiting (oczekuje na realizację)	<p>Operacja w tym stanie oczekuje na realizację, np. dla operacji typu payout oznacza zlecenie wypłaty (ręcznie przez akceptanta, bądź przez mechanizm autowypłaty).</p> <p>Operacja w tym stanie oczekuje na realizację przez n dni roboczych, gdzie n wynika z podpisanej przez akceptanta umowy.</p>
processing_realization (realizowana)	<p>Status oznacza rozpoczęcie procedury fizycznej realizacji wypłaty, dla operacji typu payout skutkuje wykonaniem przelewu na konta akceptanta, a dla operacji typu refund na konto płacącego.</p> <p>Gdy istnieje taka możliwość (np. kanały płatności kartami), zamiast wykonania przelewu, zwrot wykonywany jest na danym kanale.</p>

## 9 DZIENNIK ZMIAN

WERSJA	DATA	OPIS ZMIAN
1.94.1.0	2022-12-02	zmiana logotypu i nazwy dla kanału: 76 «Getin Bank PBL» na «VeloBank PBL» usunięcie kanału 80 «Noble Pay»
1.90.2.2	2022-05-10	zmiana logotypu dla kanału: 70 «Pocztowy24»
1.90.2.1	2022-04-27	zmiana logotypu dla kanału: 66 «Bank Nowy S.A.»
1.90.1.1	2022-04-25	zmiana logotypu dla kanału: 246 i 248 «Karty płatnicze» zmiana logotypu dla kanału: 91 «Nest Bank.»
1.87.9.1	2022-01-05	zmiana logotypu i nazwy dla kanału: 66 «Bank Nowy S.A.»
1.86.1.1	2021-11-20	usunięcie kanału 81 «Idea Cloud»
1.84.3.1	2021-09-24	zmiana limitu dla kanału 95 «PayPo»
1.84.2.1	2021-09-17	usunięcie kanału 83 «EnveloBank»

ciąg dalszy na następnej stronie

Table 28 - kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.81.8.1	2021-08-03	dodanie nowych <i>adresów IP</i> z których Dotpay może przysyłać powiadomienia do serwisów sprzedawców
1.79.20.1	2021-05-25	dodanie nowej wersji api <code>api_version : next</code> <b>Zmiana sposobu liczenia sumy kontrolnej</b> służącej do weryfikacji poprawności przesłanych danych - <i>Ochrona integralności parametrów przekierowania (CHK) : chk</i> zmiana treści wyświetlonych informacji przy wykorzystaniu parametrów <code>bylaw</code> i <code>personal_data</code>
1.78.22.1	2021-04-26	dodanie dodatkowych informacji w Rozdziale <i>Odbiór potwierdzenia o zaksięgowanej wpłacie w serwisie Dotpay</i> aktualizacja listy dostępnych kanałów płatności
1.77.10.1	2021-02-22	zmiana logotypu dla kanału: 71 «MasterPass», 246 «Karty płatnicze», 248 «Karty płatnicze» usunięcie kanału 35 «Kantor Polski»
1.75.7.1	2020-11-30	zmiana logotypu dla kanału: 1 «mTransfer» dodanie nowych wartości dla parametru <code>lang = lt</code> (język litewski) i <code>lang = lv</code> (język łotewski) usunięcie kanału 60 «Płacę z T-Mobile Usługi Bankowe»
1.74.2.1	2020-11-16	dodanie dodatkowego opcjonalnego parametru do <i>notyfikacji URLC</i> : <code>operation_seller_code</code> dodanie nowej grupy kanałów w <code>channel_groups</code>
1.73.13.1	2020-10-16	zmiana logotypu dla kanału: 50 «Pay Way Toyota Bank»
1.72.3.1	2020-09-15	usunięcie kanału 21 «VIA - Moje Rachunki» dodanie nowych parametrów <code>order.delivery_address.name</code> , <code>order.delivery_address.phone</code> , <code>order.delivery_address.is_verified</code> w Rozdziale <i>Obsługa danych dostawy oraz płacącego</i> dodanie wymagań dla danych przesyłane w parametrze <code>customer</code> dla kanału <i>PayPal - tylko dla SPP (Seller Protection Policy)</i>
1.71.10.1	2020-08-18	usunięcie kanału 84 «Volkswagen Bank direct» dodanie dodatkowego opcjonalnego parametru do <i>notyfikacji URLC</i> : <code>channel_reference_id</code>
1.70.0.1	2020-07-27	zmiana logotypu dla kanału: 55 «Raty z Alior Bankiem» oraz 249 «Visa SRC»
1.69.18.2	2020-06-26	zmiana nazwy i logotypu dla kanału: 249 «Visa SRC» (wcześniej: «Visa Checkout»)
1.69.18.1	2020-06-19	zmiana logotypu dla kanału: 94 «Kupuj teraz zapłać później»
1.67.18.2	2020-04-22	usunięcie kanału 65 (Płacę z Idea Bank)
1.67.18.1	2020-04-22	dodanie nowego parametru <code>order.items.category</code> w Rozdziale <i>Obsługa danych dostawy oraz płacącego</i> dodanie wymagań dla danych przesyłane w parametrze <code>customer</code> dla kanału <i>Raty Alior</i> dodanie dodatkowych opcjonalnych parametrów do <i>notyfikacji URLC</i> : <code>payer_bank_account_name</code> , <code>payer_bank_account</code> , <code>payer_transfer_title</code> , <code>blik_voucher_pin</code> , <code>blik_voucher_amount</code> , <code>blik_voucher_amount_used</code>
1.65.2.1	2020-02-11	zmiana nazwy i logotypu dla kanału: 66 «Bank Nowy BFG S.A.» (wcześniej: «Płacę z PBS»)

ciąg dalszy na następnej stronie

Table 28 - kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.62.2.1	2019-11-08	usunięcie kanałów 48 (BNP Paribas - Płacę z Żółty), 88 (BNP Paribas dawni Klienci Raiffeisen), 56 (eurobank - płatność on-line) dodanie kanału 262 (Apple Pay) zmiana logotypu dla kanałów: 90 (BNP Paribas - płacę z Pl@net) rozszerzenie parametru customer o dodatkowe pola w Rozdziale <i>Obsługa danych dostawy oraz płacącego</i> ( <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> ) dodanie parametru <i>ap_token</i>
1.60.16.1	2019-09-19	usunięcie kanału 33 (Volkswagen Bank)
1.59.10.2	2019-08-12	dodanie nowego parametru <i>order.id</i> w Rozdziale <i>Obsługa danych dostawy oraz płacącego</i> , dodanie opisu parametru <i>pid</i> poprawki typograficzne
1.59.10.1	2019-07-12	zmiana logotypu dla kanału 51 (Płać z BOŚ) oraz kanału 95 (PayPo)
1.58.2.1	2019-06-10	zmiana nazwy i logotypu dla kanału: 55 «Raty z Alior Bankiem» (wcześniej: «erata - raty z dotpay»)
1.58.0.2	2019-06-03	zmiana nazwy i logotypu dla kanału: 93 «Kasa Stefczyka» (wcześniej: «eSKOK») dodanie nowego Rozdziału: <i>Bezpieczeństwo integracji płatności</i> poprawki typograficzne
1.58.0.1	2019-05-23	poprawki typograficzne
1.56.14.1	2019-04-24	dodanie nowej wartości dla parametru <i>currency</i> : BGN, CHF, HRK, HUF, RUB
1.56.11.3	2019-04-10	usunięcie kanału 72 (Płacę z Orange) dodanie parametru <i>gp_token</i>
1.56.11.2	2019-04-01	zmiana nazw oraz logotypów dla kanałów: 32 (dawny: BGŻ BNP Paribas), 48 (dawny: R - Przelew), 88 (dawny: Raiffeisen), 90 (dawny: BGŻ BNP Paribas) dodanie parametru <i>customer</i> oraz związanego z nim Rozdziału <i>Obsługa danych dostawy oraz płacącego</i>
1.56.11.1	2019-03-28	dodanie kanału 95 (PayPo) dodanie kanału 260 (Google Pay) dodanie aktualnej listy kanałów do pobrania w formacie <i>xlsx</i> zmiana nazw oraz logotypów dla kanałów: 32 (dawny: BGŻ BNP Paribas), 48 (dawny: R - Przelew), 88 (dawny: Raiffeisen), 90 (dawny: BGŻ BNP Paribas)
1.55.8.1	2019-03-22	usunięcie kanału 31 (Zapłać w Żabce i we Freshmarket) usunięcie kanału 24 (mPay)
1.55.7.1	2019-02-15	dodanie nowej wartości dla parametru <i>operation_type</i> = <i>pay-out_commission</i> dodanie nowej wartości dla parametru <i>lang</i> = <i>uk</i> (język ukraiński) dodanie nowej wartości dla parametru <i>currency</i> = <i>NOK</i> wykluczenie z dokumentacji alternatywnych nazw parametrów przesyłanych do serwisu Dotpay: <i>kwota</i> , <i>waluta</i> , <i>opis</i> , <i>kanal</i> , <i>blokuje</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>

ciąg dalszy na następnej stronie

Table 28 - kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.53.5.1	2018-12-10	poprawki typograficzne dodanie parametru <code>credit_card_unique_identif</code> w notyfikacjach <i>URLC</i>
1.52.6.1	2018-11-13	usunięcie kanału 58 (Szybkie Płatności Internetowe z Deutsche Bank PBC) dodanie wyrażenia regularnego dla parametru <code>amount</code>
1.51.0.1	2018-10-03	dodanie parametrów <code>credit_card_expiration_year</code> i <code>credit_card_expiration_month</code> w notyfikacjach <i>URLC</i>
1.50.11.1	2018-09-27	dodanie kanału 59 (CinkciarzPAY) zmiana nazwy dostawcy kanału z Volkswagen Bank Polska S.A. na Volkswagen Bank GmbH
1.50.8.2	2018-09-07	zmiana logotypu dla kanału 6 (Przelew24) oraz kanału 89 (BZWBK)
1.50.8.1	2018-09-05	Włączenie wymagalności przesyłania parametru <code>chk</code> dla nowo powstałych kont w systemie Dotpay zmiany w opisach dotyczące konieczności weryfikacji kwoty i waluty zamówienia
1.49.11.1	2018-06-26	dodanie wartości O - płatności odroczone M - płatności mobilne (DCB) dla parametru <code>channel_groups</code> (grupy kanałów) dodanie kanału 231 (Orange) dodanie kanału 232 (T-Mobile) dodanie kanału 233 (PLAY) dodanie kanału 234 (Plus) dodanie kanału 94 (Kupuj teraz, zapłać później) dodano informację o TLSv1.2
1.45.2.1	2018-03-30	zmiana nazwy podrozdziału Płatności one-click na One-click i płatności cykliczne dodanie nowych parametrów do powyższego rozdziału, oraz uzupełnienie ich w sekcji Ochrona integralności parametrów przekierowania (CHK) dodanie nowych wartości dla parametru <code>error_code</code> opisanego w rozdziale Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy
1.44.12.2	2018-03-08	dodanie kanału 93 (eSKOK)
1.44.12.1	2018-02-20	zmiana logotypu dla kanału 48 (R-Przelew) oraz 246, 248 (Karty płatnicze)
1.44.10	2018-02-13	dodanie nowych wartości dla parametru <code>currency</code> (waluta) dodanie nowych wartości dla parametru <code>language</code> (język) dodanie przykładowych kart do rozdziału ŚRODOWISKO TESTOWE
1.39.2.1	2017-09-03	dodanie kanału 83 (EnveloBank) dodanie kanału 249 (Visa Checkout) zmiana logotypu dla kanału 246 (Karty płatnicze via Payeezy) oraz 248 (Karty płatnicze)
1.38.1.1	2017-08-21	dodanie kanału 15 (iPKO)
1.37.3.3	2017-07-10	dodanie parametru <code>is_completed</code> w notyfikacjach <i>URLC</i>
1.37.3.2	2017-06-28	dodanie podrozdziału Masscollect

ciąg dalszy na następnej stronie

Table 28 - kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.37.3.1	2017-06-27	dodanie podrozdziału Płatność dzielona (Multimerchant / Pasaż) dodanie parametrów z funkcjonalności Multimerchant oraz Surcharge do podrozdziału Ochrona integralności parametrów przekierowania (CHK) aktualizacja podrozdziału Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy dodanie parametru ignore_last_payment_channel
1.36.10.1	2017-06-12	zmiana logotypu dla kanału 36 (Pekao24Przelew)
1.36.7.1	2017-06-01	dodanie kanału 90 (BGŻ BNP Paribas) dodanie kanału 91 (Nest Bank) dodanie kanału 92 (Bank Spółdzielczy w Brodnicy) dodanie informacji o możliwościach pobrania listy dostępnych kanałów (channel) dla konkretnego sklepu
1.35.4.2	2017-04-05	dodanie wartości UAH dla parametru currency (waluta) dodanie kanału 88 (Raiffeisen) dodanie kanału 89 (BZWBK)
1.35.4.1	2017-03-28	usunięcie kanału 18 (Przelew z BPH)
1.34.9.3	2017-01-26	dodanie parametru credit_card_registration w notyfikacjach URLC
1.34.9.2	2017-01-11	dodanie wyrażeń regularnych zmiana logotypów dla kanałów: 71 (MasterPass), 246 (Karty płatnicze via Payeezy), 248 (Karty płatnicze)
1.34.9.1	2017-01-02	usunięcie kanału 77 (FerBuy) usunięcie wartości O - płatności odroczone dla parametru channel_groups (grupykanałów)
1.33.4.2	2016-12-06	dodanie kanału 218 (paysafecard)
1.33.4.1	2016-11-14	usunięcie kanału 27 (BGŻ)
1.32.6.2	2016-11-07	dodanie parametru credit_card_registration usunięcie kanału 63 (Płacę z IKO)
1.32.6.1	2016-09-23	dodanie kanału 84 (Volkswagen Bank direct) dodanie kanału 86 (TrustPay) dodanie kanału 87 (Credit Agricole PBL)
1.30.6.3	2016-06-22	dodanie parametru deladdr
1.30.6.2	2016-06-17	poprawki typograficzne dodanie parametrów bylaw, personal_data dodanie parametrów credit_card_number, credit_card_expiration_date_year, credit_card_expiration_date_month, credit_card_security_code zmiana sposobu pozyskiwania dostępu do środowiska testowego (dodanie odnośnika do rejestracji)
1.30.6.1	2016-06-01	poprawki typograficzne dodanie podrozdziału Płatność one-click dodanie parametru expiration_date dodanie wartości HASH_NOT_EQUAL_CHK dla parametru error_code opisanego w rozdziale Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy
1.29.11.1	2016-03-21	dodanie rozdziału DODATKOWE FUNKCJONALNOŚCI dodanie podrozdziału Ochrona integralności parametrów przekierowania (CHK) przeniesienie podrozdziału Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy do rozdziału DODATKOWE FUNKCJONALNOŚCI

ciąg dalszy na następnej stronie

Table 28 - kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.29.8.1	2016-02-26	dodanie parametru <code>channel_groups</code> dodanie informacji o <i>HTTPS verify</i> oraz <i>SSL certificate verify</i> w rozdziale II. ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA <i>URLC</i> ) dodanie kanału 82 (Przelew SEPA) dodanie kanału 248 (Karty płatnicze) usunięcie kanału 64 (PeoPay)
1.28.5.2	2016-01-07	dodanie parametru <code>blík_code</code> dodanie rozdziału ZAŁĄCZNIK II (OPISY STATUSÓW OPERACJI)
1.28.5.1	2015-12-31	usunięcie kanału 79 (Open Pay)
1.27.0.1	2015-11-10	usunięcie kanału 25 (Plus Bank)
1.25.7.1	2015-10-30	usunięcie kanału 49 (MeritumBank)
1.25.3.1	2015-10-22	usunięcie kanału 22 (Ukash) usunięcie kanału 43 (Bank Spółdzielczy we Wschowie) usunięcie kanału 62 (DNB Nord)
1.25.1.1	2015-10-19	zmiana adresu produkcyjnej strony płatności (z <a href="https://ssl.dotpay.pl">https://ssl.dotpay.pl</a> na <a href="https://ssl.dotpay.pl/t2/">https://ssl.dotpay.pl/t2/</a> ) usunięcie kanału 3 (MultiTransfer)
1.24.9.1	2015-10-12	dodanie kanału 81 (Idea Cloud)
1.23.13.3	2015-09-24	poprawki typograficzne
1.23.13.2	2015-08-20	usunięcie kanału 69 (V.me)
1.23.13.1	2015-08-12	dodanie kanału 79 (Open Pay) dodanie kanału 80 (Noble Pay) usunięcie kanału 15 (iPKO)
1.23.9.2	2015-07-30	usunięcie wartości <code>UNKNOWN_ACCOUNT</code> dla parametru <code>error_code</code> opisanego w rozdziale Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy
1.23.9.1	2015-07-20	dodanie kanału 77 (FerBuy)
1.22.9.1	2015-06-01	dodanie kanału 74 (Banki Spółdzielcze) dodanie kanału 75 (Płacę z Plus Bank) dodanie kanału 76 (Getin Bank PBL) usunięcie kanału 17 (Płacę z iPKONET) usunięcie kanału 57 (Getin Bank) dodanie parametrów <code>operation_withdrawal_amount</code> , <code>operation_commission_amount</code> , <code>channel_country</code> , <code>geoip_country</code> w notyfikacjach <i>URLC</i>
1.20.9.2	2015-02-09	dodanie kanału 73 (BLIK)
1.20.9.1	2015-01-14	usunięcie kanału 245 (MasterCard Mobile) dodanie rozdziału DZIENNIK ZMIAN.
1.19.15.2	2014-12-12	dodanie kanału 72 (Płacę z Orange)
1.19.15.1	2014-12-08	dodanie kanału 71 (MasterPass)
1.18.5.4	2014-11-04	dodanie kanału 69 (V.me)
1.18.5.3	2014-11-19	usunięcie kanałów 14 (KB24) oraz 61 (Bank Pocztowny)
1.18.5.2	2014-10-28	dodanie kanałów 66 (Płacę z PBS ) oraz 70 (Pocztowny24) zmiana nazwy kanału 17 (z Płać z Nordea na Płać z IPKOnet) dodanie adresu <a href="https://ssl.dotpay.pl/test_seller/">https://ssl.dotpay.pl/test_seller/</a> do rozdziału ŚRODOWISKO TESTOWE

## HTTP Routing Table

/cards

DELETE /cards/(string:credit\_card\_id)/,  
60