Dotpay
Instrukcja techniczna implementacji płatności

wydanie: 1.62.2.1 pl

Dotpay Development Team

Utworzono : 11 lutego 2020

Ta dokumentacja dostępna jest również online

w formacie html

Contents

1 Wersja podstawowa **1**
1.1 Przykładowe formularze płatności / dotacji . . . 2

2 Wersja zaawansowana **6**
2.1 Diagram 1. Przykładowy przebieg procesu płatności przedstawia poniższy schemat oraz opis: . . . 7
2.2 Tabela 1. (Podstawowe parametry przesyłane do serwisu Dotpay) . . . 8
2.3 Tabela 2. (Dodatkowe parametry przesyłane do serwisu Dotpay) . . . 9

3 ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC) **20**
3.1 Tabela 3. (Parametry wysyłane przez serwis Dotpay po wykonaniu operacji (transakcji)) . . . 22

4 DODATKOWE FUNKCJONALNOŚCI **29**
4.1 Ochrona integralności parametrów przekierowania (CHK) . . . 29
4.1.1 Przykład 1 podstawowy . . . 30
4.1.2 Przykład 2 pełny . . . 34
4.2 One-click i płatności cykliczne . . . 40
4.2.1 Tabela 4. (Parametry związane z rejestracją karty płatniczej) . . . 41
4.2.2 Tabela 5. (Parametry wykorzystywane w procesie kolejnej płatności zarejestrowaną wcześniej kartą) . . . 41
4.2.3 Tabela 6. (Parametry wspólne dla pierwszej oraz kolejnej płatności) . . . 42
4.2.4 Przykładowe modele integracji oraz związane z nimi podstawowe wymagania) . . . 43
Model integracji: 1 click . . . 43
Model integracji: recurring . . . 43
Model integracji mieszany: 1 click + recurring . . . 44
4.2.5 Tabela 11. (Parametry dla danych karty płatniczej) . . . 46
4.3 Płatność dzielona (Multimerchant / Pasaż) . . . 48
4.4 Masscollect . . . 49
4.4.1 Tabela 12. (Parametry dodatkow wykorzystywane w usłudze Masscollect) 50
4.5 Obsługa danych dostawy oraz płacącego . . . 51
4.5.1 Tabela 13. (Dane obsługiwane przez parametr `customer`) . . . 52
4.5.2 Wartości używane dla pola zastępczego typu indykator dla wybranych pól: . . . 54
4.5.3 Wymagane dane przesyłane w parametrze `customer` dla kanału **PayPo** 55
4.6 Obsługa błędnych przekierowań przesyłanych z systemu sprzedawcy . . . 57

5 ŚRODOWISKO TESTOWE **58**
5.1 Tabela 14. (Lista przykładowych numerów kart, które można wykorzystać w procesie testowania płatności kartowych) . . . 58

6 INFORMACJE DODATKOWE **59**
6.1 Weryfikacja adresu IP . . . 59
6.2 Bezpieczeństwo integracji płatności . . . 59
6.2.1 Przekierowanie kupującego do Dotpay . . . 59
6.2.2 Odbiór potwierdzenia o zaksięgowanej wpłacie w serwisie Dotpay . . . 60

7 ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI) **60**
7.1 Tabela 15. (Kanały płatności dostępne w serwisie Dotpay) . . . 60
7.1.1 KARTY PŁATNICZE . . . 60
7.1.2 SZYBKIE TRANSFERY . . . 61
7.1.3 PRZELEWY ONLINE . . . 64
7.1.4 PŁATNOŚCI GOTÓWKOWE . . . 64
7.1.5 PORTMONETKI I VOUCHERY . . . 65
7.1.6 RATY . . . 65

7.1.7 INNE 66
7.1.8 PŁATNOŚCI ODROCZONE 66
7.1.9 PŁATNOŚCI MOBILNE typu Direct Carrier Billing 67
7.2 Kanały płatności - do pobrania 68

8 ZAŁĄCZNIK II (OPISY STATUSÓW OPERACJI) 68
8.1 Tabela 16. (Statusy operacji tworzone w systemie Dotpay) 68

9 DZIENNIK ZMIAN 69

HTTP Routing Table 74

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/*

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: rfhu4jb5ym657g3xluf4bbqfmbyj6t17

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

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

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=rfhu4jb5ym657g3xluf4bbqfmbyj6t17&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=rfhu4jb5ym657g3xluf4bbqfmbyj6t17&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

---

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[1]

```
<html>
<!------------------------------

        Górna część strony

-------------------------------------->

<script type="text/javascript" src="http://code.jquery.com/jquery-3.3.1.min.js"></
↪script>

<script>
  $(document).ready(function () {
         $("#buttomDarowizna").click(function () {
             if($('#kwota').val().trim() === '') {
                 $("#kwota_alert").text("Proszę wybrać lub wprowadzić kwotę␣
↪darowizny.").show();
                 $('#kwota_alert').css( "display", "inline" ).fadeOut( 5000 );
                 return false;
             }
         });
      });
</script>

<div align="center">
  <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
    <p style="font-size: 18px">Wybierz kwotę darowizny</p>
    <p>
      <input type="button" onClick="$('#kwota').val('10'); $('#kwota').prop(
↪'disabled', true);" value="10.00 zł" />
      <input type="button" onClick="$('#kwota').val('20'); $('#kwota').prop(
↪'disabled', true);" value="20.00 zł" />
      <input type="button" onClick="$('#kwota').val('50'); $('#kwota').prop(
↪'disabled', true);" value="50.00 zł" />
      <input type="hidden" name="type" value="0" />
      <input type="hidden" name="currency" value="PLN" />

      <table>
        <tr>
          <td><input type="button" onClick="$('#kwota').prop('disabled', false);"␣
↪value="Inna kwota" /></td>
          <td>Kwota:<br /><input type="text" name="amount" id="kwota" size="5"␣
↪disabled="disabled" pattern="^([1-9])((\.\d{1,2})?)$|^((?!0)(\d){1,5})((\.\d{1,
↪2})?)$|^(1(\d{5})(.\d{1,2})?)$|^(200000(.[0]{1,2})?)$" placeholder="np. 100"␣
↪maxlength="9"
```

(ciąg dalszy na następnej stronie)

[1] https://www.dotpay.pl/developer/doc/api_payment/pl/files/code/form/form1.html

(kontynuacja poprzedniej strony)

```
              size="9" title="Kwota powinna zawierać się w przedziale 1 - 200000␣
↪PLN. Dozwolony format to np: 100 lub 152.43" /></td>
        </tr>
      </table>

      <!--------------------------------- KONFIGURACJA ----------------------------
↪----------->

      <!---- zamiast 000000 nalezy podstawic numer ID w Dotpay -->
      <input type="hidden" name="id" value="000000" />
      <!--- Tytuł transakcji --->
      <input type="hidden" name="description" value="Testowa płatność" />
      <!--- Ardes URL powrotu --->
      <input type="hidden" name="url" value="http://www.example.com" />
      <!-- Tekst przycisku powrotu do sklepu --->
      <input type="hidden" name="buttontext" id="buttontext" value="Powrót do␣
↪sprzedawcy" />
    </p>

    <!--------------------------------- KONIEC KONFIGURACJI ----------------------
↪---------------->

    <p><br><button class="buttomDarowizna" id="buttomDarowizna">Wpłać darowiznę</
↪button></p>
  </form>
  <div id="kwota_alert" style="color:red;"></div>
</div>

<!------------------------------

        Dolna część strony

------------------------------------->

</html>
```

2. Formularz wyłącznie z predefiniowanymi kwotami[2]

```
<html>
<!------------------------------

        Górna część strony

------------------------------------->


<div align="center">
  <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
    <p style="font-size: 18px">Wybierz kwotę darowizny</p>
    <input type="radio" name="amount" value="10.00" />10.00 zł   
    <input type="radio" name="amount" value="20.00" />20.00 zł   
    <input type="radio" name="amount" value="50.00" checked />50.00 zł<br />
    <input type="hidden" name="type" value="0" />
    <input type="hidden" name="currency" value="PLN" />

    <!--------------------------------- KONFIGURACJA ----------------------------
↪--------->

    <!---- zamiast 000000 nalezy podstawic numer ID w Dotpay -->
```

(ciąg dalszy na następnej stronie)

[2] https://www.dotpay.pl/developer/doc/api_payment/pl/files/code/form/form2.html

(kontynuacja poprzedniej strony)

```
    <input type="hidden" name="id" value="000000" />
    <!--- Tytuł transakcji --->
    <input type="hidden" name="description" value="Testowa płatność" />
    <!--- Ardes URL powrotu do sklepu--->
    <input type="hidden" name="url" value="http://www.example.com" />
    <!-- Tekst przycisku powrotu do sklepu --->
    <input type="hidden" name="buttontext" value="Powrót do sprzedawcy" />

    <!---------------------------------- KONIEC KONFIGURACJI ----------------------
↪---------------->

    <p><br><button class="buttomDarowizna">Wpłać darowiznę</button></p>
  </form>
</div>

<!-----------------------------

        Dolna część strony

------------------------------------->

</html>
```

3. Formularz z dowolnym opisem i kwotą[3]

```
<html>
<meta charset="utf-8">
<!-----------------------------

        Górna część strony

------------------------------------->

<div align="center">
  <form action="https://ssl.dotpay.pl/t2/" method="post" target="_parent">
    <p style="font-size: 18px">Wybierz kwotę darowizny</p>
    <input name="description" value="Darowizna na cele statutowe" type="hidden">
    <input name="ignore_last_payment_channel" value="1" type="hidden">
    <input name="amount" id="kwota" size="6" value="" type="text" required
      pattern="^([1-9])((\.\d{1,2})?)$|^((?!0)(\d){1,5})((\.\d{1,2})?)$|^(1(\d{5}
↪)(.\d{1,2})?)$|^(200000(.[0]{1,2})?)$"
      placeholder="np. 10" maxlength="9" size="9"
      title="Kwota powinna mieścić się w przedziale 1 - 200 000 PLN. Dozwolony␣
↪format to np: 10 lub 10.00">PLN

    <!---------------------------------- KONFIGURACJA -----------------------------
↪--------->

    <!---- zamiast 000000 nalezy podstawic numer ID w Dotpay -->
    <input name="id" value="000000" type="hidden">

    <!---------------------------------- KONIEC KONFIGURACJI ----------------------
↪----------->

    <p><br><button class="buttomDarowizna">Wpłać darowiznę</button></p>
  </form>
  <br>
</div>

```

(ciąg dalszy na następnej stronie)

[3] https://www.dotpay.pl/developer/doc/api_payment/pl/files/code/form/form3.html

(kontynuacja poprzedniej strony)

```
<!----------------------------

        Dolna część strony

------------------------------------>

</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/*.

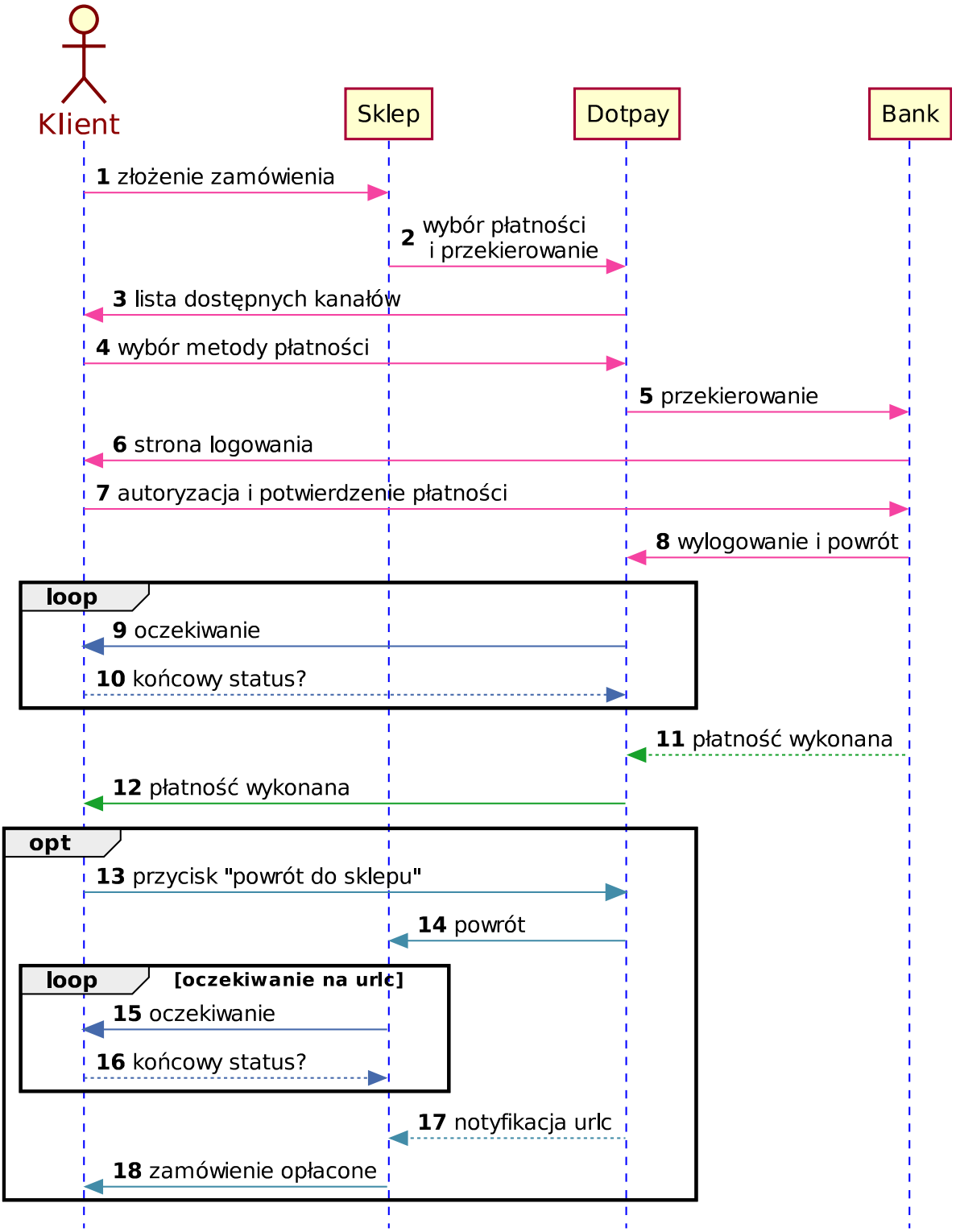
W następnym kroku klient dokonuje wpłaty wybranym kanałem płatności. Jeśli został zdefiniowany adres powrotu, oraz odpowiednia wartość parametru `type` , na stronie z potwierdzeniem płatności wyświetlony zostanie przycisk umożliwiający powrót do serwisu sprzedawcy.

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 płacącemu.

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`	Parametr określający wersję API, zgodnie z którą system prześle powiadomienie *URLC* (opis powiadomień *URLC* zawarty jest w rozdz. *ODBIERANIE INFORMACJI PO PŁATNOŚCIACH (POWIADOMIENIA URLC)*). Przesłanie parametru nadpisuje domyślną konfigurację sklepu (`id`) ustawioną w panelu administracyjnym sprzedawcy, w zakładce *Ustawienia –> Konfiguracja sklepu –> Edycja*. Dostępne wartości: dev - wersja API opisana w niniejszej instrukcji. --- **Ważne:** Jeżeli konfiguracja sklepu (*Ustawienia –> Konfiguracja sklepu –> Edycja*) wskazuje na wersję API inną niż dev, brak przesłania parametru spowoduje wysłanie przez system Dotpay nierozwijanej już wersji notyfikacji *URLC*, niezgodnej z opisem niniejszej instrukcji. --- Przykład: `api_version` = dev

ciąg dalszy na następnej stronie

Table 1 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`id`	ID sklepu sprzedawcy w systemie Dotpay, na rzecz, którego dokonywana jest płatność. Numer ID można znaleźć po zalogowaniu do panelu administracyjnego w zakładce *Ustawienia*, jest to 6-cyfrowa liczba umieszczona po znaku # w kolumnie *Sklep*. typ: *integer* minimalna wartość: *1* maksymalna wartość: *999999* Przykład: `id` = 123456
`amount`	Kwota transakcji podana z częścią setną (zawsze dwa miejsca po separatorze). Separatorem części setnej jest znak kropki. typ: *string* minimalna długość: *1* maksymalna długość: *10* przykładowe wyrażenie regularne (dla kwoty w zakresie 0.01 – 200000.00): `^0\.(0)([1-9])$\|^0\.(([1-9])(\d)?)$|^([1-9])((\.\d{1,2})?)$\|^((?!0)(\d){1,5})((\.\d{1,2})?)$|^(1(\d{5})(.\d{1,2})?)$\|^(200000(.[0]{1,2})?)$` Przykład: `amount` = 42.82
`currency`	Waluta określająca parametr `amount` , format zgodny ze standardem ISO 4217[4] . Dostępne wartości: PLN, EUR, USD, GBP, JPY, CZK, SEK, UAH, RON, NOK, BGN, CHF, HRK, HUF, RUB Przykład: `currency` = EUR
`description`	Opis przeprowadzanej operacji (transakcji). typ: *string* minimalna długość: *1* maksymalna długość: *255* Przykład: `description` = Zamówienie nr 120/2018
`chk`	Suma kontrolna służąca do weryfikacji poprawności przesłanych danych. Opis funkcjonalności znajduje się w rozdziale *Ochrona integralności parametrów przekierowania (CHK)*. **Ważne:** Parametr domyślnie wymagany.

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

[4] https://pl.wikipedia.org/wiki/ISO_4217

PARAMETR	ZNACZENIE / OPIS
`channel`	Kanał płatności, jaki ma zostać zaznaczony po przeniesieniu klienta na strony serwisu Dotpay. Dostępne wyłącznie wartości liczbowe, które zostały przedstawione w sekcji *ZAŁĄCZNIK I (KANAŁY PŁATNOŚCI)* Przykład: `channel` = 1 Po przesłaniu wartości 1 zostanie zaznaczony kanał mTransfer. Lista dostępnych kanałów płatności dla wybranego sklepu (`id`) może być pobrana za pośrednictwem: • API panelu administracyjnego sprzedawcy[5] • API płatności (mechanizm ten wykorzystywany jest również przez gotowy widget służący do wyświetlenia dostępnych kanałów, jego opis znajduje się w panelu administracyjnym sprzedawcy, w zakładce: *Narzędzia –> Widget*) Adres zasobu: *https://ssl.dotpay.pl/t2/payment_api/channels/* Wymagane parametry: `id` , `amount` , `currency` Opcjonalne parametry: `lang` , `format` (json lub xml) Przykład: https://ssl.dotpay.pl/t2/payment_api/channels/?id=123456&amount=
`ch_lock`	Wymuszenie kanału podanego w parametrze `channel` . 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. Dostępne wartości: 0 – Kanał nie będzie wymuszany (domyślnie). 1 – Wymuszenie kanału przesłanego w parametrze `channel` . Przykład: Przesłanie w przekierowaniu odpowiednio parametrów `channel` = 1 oraz `ch_lock` = 1 uniemożliwi klientowi wybranie innego kanału płatności niż mTransfer.
`ignore_last_payment_channel`	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. Domyślnie podczas kolejnej płatności prezentowana jest ostatnio wybrana metoda, wraz z możliwością ewentualnej zmiany na inną. Dostępne wartości: 1, true – Zignorowanie ostatnio wybranej metody płatności Przykład: `ignore_last_payment_channel` = 1 `ignore_last_payment_channel` = true

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`channel_groups`	Grupa kanałów płatności, jaka ma zostać wyświetlona po przeniesieniu klienta na strony serwisu Dotpay (domyślnie prezentowane są wszystkie). W przypadku korzystania z niniejszego parametru zalecane jest użycie `ignore_last_payment_channel` opisanego powyżej. Dostępne wartości: K – karty płatnicze, T – szybkie transfery, P – przelewy online, G – płatności gotówkowe, W – portmonetki i vouchery, R – raty, M – płatności mobilne (DCB), O – płatności odroczone, I – inne. W wartości parametru może być przekazanych kilka grup, w takiej sytuacji kolejne litery należy odseparować używając przecinka. Przykład: `channel_groups` = T `channel_groups` = R,I `channel_groups` = R,I,P
`url`	Adres internetowy (*HTTP* lub *HTTPS*) na jaki ma powrócić kupujący po dokonaniu płatności. Sterowanie zachowaniem parametru `url` określa parametr `type` . typ: *string* maksymalna długość: *1000* Przykład: `url` = https://www.example.com/thanks_page.php `url` = http://195.150.9.55/thanks_page.php 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 *POST* i *GET*) z wartością OK lub FAIL, który zawiera informację o ewentualnym wystąpieniu błędów na stronach płatności. --- **Ważne:** 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` . --- Przykład: `url` = https://www.example.com/thanks_page.php?status=OK

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
type	Parametr określający metodę odwołania do serwisu sprzedawcy. Wartość parametru type ma wpływ na zachowanie parametru url . Dostępne wartości: 0 – Po dokonaniu płatności kupującemu zostanie udostępniony przycisk powrotu do serwisu sprzedawcy, 2 – Brak reakcji, bez przycisku (wartość domyślna). 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 cześć procesu płatności zostanie zrealizowana bez zmian, tj. nastąpi automatyczny powrót od dostawcy kanału (tzw. **on-site** / **white label**) --- **Ważne:** 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 Dotpay Sp. z o.o. danych osobowych płacącego dla potrzeb realizacji procesu płatności (parametr personal_data). --- Przykład: Po przesłaniu poniższego zestawu parametrów, kupującemu zostanie wyświetlony (po dokonaniu płatności) przycisk pozwalający przejść do strony *https://www.example.com/thanks_page.php*: type = 0 url = https://www.example.com/thanks_page.php 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. *Sklep –> Bank –> Sklep*: type = 4 url = https://www.example.com/thanks_page.php bylaw = 1 personal_data = 1

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`buttontext`	Treść, która zostanie wyświetlona na przycisku powrotu do serwisu sprzedawcy. Domyślną wartością jest Powrót do sklepu. typ: *string* minimalna długość: *4* maksymalna długość: *100* Przykład: `buttontext` = Wróć do www.example.com
`bylaw`	Parametr informujący o zaakceptowaniu przez klienta regulaminu płatności oraz polityki cookies Dotpay Sp. z o.o.. W przypadku korzystania z niniejszego parametru system sprzedawcy powinien wyświetlić kupującemu pole wyboru z treścią analogiczną do poniższej.. `Akceptuję <a title="regulamin płatności" target="_blank" href="https://ssl.dotpay.pl/t2/cloudfs1/magellan_media/regulamin_platnosci">Regulamin płatności</a> Dotpay sp. z o.o..` Dostępne wartości: 1 – akceptacja regulaminu płatności Przykład: `bylaw` = 1
`personal_data`	Parametr informujący o wyświetleniu klientowi informacji na temat przetwarzanie danych osobowych przez Dotpay Sp. z o.o.. W przypadku korzystania z niniejszego parametru system sprzedawcy powinien wyświetlić treść analogiczną do poniższej: `Przyjmuję do wiadomości, że w celu realizacji procesu płatności Administratorem moich danych osobowych jest Dotpay sp. z o.o. (KRS 0000700791), 30-552 Kraków (Polska), Wielicka 28B, +48126882600, bok@dotpay.pl, zobacz pełną treść klauzuli informacyjnej.` Dostępne wartości: 1 – informacja o zapoznaniu się z klauzulą informacyjną Przykład: `personal_data` = 1

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`urlc`	Adres internetowy (*HTTP* lub *HTTPS*) do odbioru parametrów potwierdzających realizację lub odmowę operacji (transakcji). Dokładny opis mechanizmu powiadomień *URLC* został zawarty w rozdz. *ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA URLC)*. Zalecane jest, aby podany adres kierował bezpośrednio do odbierającego pliku. Jeżeli po drodze nastąpi przekierowanie typu *301* lub *302* (które zezwalają na zmianę metody HTTP) otrzymana od Dotpay notyfikacja może być pusta. Ewentualnie można ustawić na serwerze przekierowanie typu *307* / *308*, które na takie zachowanie nie pozwalają. Przesłanie parametru nadpisuje domyślną konfigurację sklepu (`id`) ustawioną w panelu administracyjnym sprzedawcy, w zakładce *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja* . **Ważne:** Jeżeli w panelu nie zostało odblokowane przyjmowanie parametru `urlc` z zewnętrznych źródeł (odznaczona opcja *Blokuj zewnętrzne urlc* w menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*), to przesyłany parametr `urlc` jest ignorowany. **Ważne:** Jeżeli opcja *HTTPS verify* z zakładki *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja* jest włączona (domyślne ustawienie), przesłany adres musi korzystać z protokołu HTTPS. typ: *string* maksymalna długość: *1000* Przykład: `urlc` = https://www.example.com/urlc_receiver.php `urlc` = http://195.150.9.55/urlc_receiver.php
`expiration_date`	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 *Ochrona integralności parametrów przekierowania (CHK)* . Format: *YYYY-MM-DD hh:mm:ss* Przykład: `expiration_date` = 2019-06-01 12:06:37
`control`	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 *URLC*. typ: *string* maksymalna długość: *1000* Przykład: `control` = ec4bf09d3dbe0cb71e6abc3ea44a7273

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

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

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`state`	Dodatkowy parametr adresu lub stan/region. typ: *string* maksymalna długość: *50* Przykład: `state` = NY
`addr3`	Dodatkowy parametr adresu. typ: *string* maksymalna długość: *50*
`city`	Nazwa miejscowości. typ: *string* minimalna długość: *1* maksymalna długość: *50* wyrażenie regularne: `^[\p{L}0-9\.\s\-\/\'_,]{1,50}$` Przykład: `city` = Kraków
`postcode`	Kod pocztowy. typ: *string* maksymalna długość: *20* wyrażenie regularne: `^[\d\w\s\-]{0,20}$` Przykład: `postcode` = 30-552
`phone`	Numer telefonu. typ: *string* minimalna długość: *3* maksymalna długość: *20* wyrażenie regularne: `^[\+\s0-9\-_]{3,20}$` Przykład: `phone` = +48 127654321
`country`	Nazwa kraju, z którego pochodzi osoba dokonująca płatność. Format zgodny ze standardem ISO 3166-1[6] (alfa-2 lub alfa-3) lub tekstowa skrócona angielska nazwa kraju. typ: *string* maksymalna długość: *50* Przykład: `country` = PL `country` = POL `country` = Poland

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`lang`	Język prezentowanych stron i formularzy dokonywania płatności. Brak przesłania parametru spowoduje wyświetlenie stron w języku przeglądarki płacącego. Jeżeli dany język nie jest dostępny, zostanie użyty angielski. Dostępne wartości: cs – język czeski, de – język niemiecki, en – język angielski, es – język hiszpański, fr – język francuski, hu – język węgierski, it – język włoski, pl – język polski, ro – język rumuński, ru – język rosyjski, uk – język ukraiński Przykład: `lang` = en
`customer`	Dane odbiorcy oraz dostawy. Parametr wymagany w przypadku niektórych kanałów płatności (np. dla kanału 95 - "PayPo"). Dane te powinny zostać zakodowane do formatu JSON[7] a następnie kodowane przy użyciu Base64[8]. Lista obsługiwanych danych dla tego parametru oraz sposób ich kodowania został zaprezentowany w Rozdziale *Obsługa danych dostawy oraz płacącego*. Wartość nie jest zwracana do sprzedawcy (np. w notyfikacjach *URLC* lub panelu administracyjnym). --- **Ważne:** Do obsługi tego parametru dane konto (`id`) musi posiadać wymuszoną weryfikację *CHK* po stronie Dotpay. --- typ: *string* Przykład: `customer` = eyJyZWdpc3RlcmVkX3NpbmNlIjoiMjAxNy0xMi0zM
`deladdr`	Adres dostawy. Parametr pełni jedynie funkcję informacyjną dla systemu Dotpay. Wartość nie jest zwracana do sprzedawcy (np. w notyfikacjach *URLC* lub panelu administracyjnym). typ: *string* maksymalna długość: *500* Przykład: `deladdr` = Punkt odbioru 3, Warszawa, ul. Ogonowa 14

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`p_info`	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. Przesłanie parametru nadpisuje domyślną konfigurację sklepu (`id`) ustawioną w panelu administracyjnym sprzedawcy, w zakładce *Ustawienia –> Konfiguracja sklepu –> Edycja.* typ: *string* maksymalna długość: *300* Przykład: `p_info` = Sklep example.com
`p_email`	Adres e-mail, który zostanie wyświetlony kupującemu w celu kontaktu ze sprzedawcą. Przesłanie parametru nadpisuje domyślny adres sklepu podany podczas rejestracji w serwisie Dotpay. Przesłanie parametru nadpisuje domyślną konfigurację sklepu (`id`) ustawioną w panelu administracyjnym sprzedawcy, w zakładce *Ustawienia –> Konfiguracja sklepu –> Edycja*. typ: *string* maksymalna długość: *100* Przykład: `p_email` = biuro@example.com

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`pid`	Link płatniczy dla danego sklepu (`id`)generowany za pośrednictwem Panelu administracyjnego Dotpay (w zakładce *Narzędzia –> Generator linków płatniczych –> Generuj link*) lub za pośrednictwem API panelu administarcyjnego[9]. typ: *string* długość: *32* Przykład: `pid` = rfhu4jb5ym657g3xluf4bbqfmbyj6t17 Parametr `pid` może występować albo samodzielnie w linku płatniczym, np: https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmbyj6t1 albo z dodatkowymi parametrami: `lang` , `ignore_last_payment_channel` , np: https://ssl.dotpay.pl/t2/?pid=rfhu4jb5ym657g3xluf4bbqfmb **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.

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`blik_code`	Kod BLIK, zatwierdzający płatność niniejszym kanałem. W standardowym procesie kod jest podawany przez klienta na stronie dostawcy płatności, po wybraniu kanału w serwisie Dotpay. Przekazanie parametru pozwala skrócić ścieżkę płatności, gdyż kod może być podawany przez klienta już na stronie sklepu sprzedawcy wraz z innymi danymi zamówienia. typ: *string* minimalna długość: *6* maksymalna długość: *6* wyrażenie regularne: `^[\d]{6}$` Przykład: `blik_code` = 264230
`gp_token`	Zakodowany przy użyciu funkcji **Base64** token otrzymany z Google Pay po wdrożeniu przez Akceptanta Google Pay API[10]. Zawiera on zaszyfrowane dane tokenizowanej karty płacącego służące do realizacji płatności. Przekazanie parametru pozwala skrócić ścieżkę płatności, ponieważ autentykacja danych kartowych płacącego następuje już na stronie sklepu sprzedawcy. typ: *string* Przykład: `gp_token` = eyJzaWduYXR1cmUiOiJNRVFDSUZDSm5MQWI1Rk50N3
`ap_token`	Zakodowany przy użyciu funkcji **Base64** token otrzymany z Apple Pay po wdrożeniu przez Akceptanta Apple Pay. Zawiera on zaszyfrowane dane tokenizowanej karty płacącego służące do realizacji płatności. Przekazanie parametru pozwala skrócić ścieżkę płatności, ponieważ autentykacja danych kartowych płacącego następuje już na stronie sklepu sprzedawcy/urządzeniu płacącego. typ: *string* Przykład: `ap_token` = eyJzacmUiOiJ … h0UXAwM2RcIn0ifQ==

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*

[5] https://www.dotpay.pl/developer/doc/api_seller/pl/#lista-kanalow-platnosci-sklepu-id
[6] https://pl.wikipedia.org/wiki/ISO_3166-1
[7] https://en.wikipedia.org/wiki/JSON
[8] https://pl.wikipedia.org/wiki/Base64
[9] https://www.dotpay.pl/developer/doc/api_seller/pl/#tworzenie-linku-platniczego
[10] 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 kilkukrotnie 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 identyfikowana 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) 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 .

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: *integer* minimalna wartość: *1* maksymalna wartość: *999999* Przykład: `id` = 123456
`operation_number`	Numer operacji (transakcji). Format numeru opisuje wyrażenie regularne: `^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)* --- **Informacja:** 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: *string* 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[11] . 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: *string* minimalna długość: *1* maksymalna długość: *10* Przykład: `operation_original_amount` = 42.82 --- **Ważne:** Brak weryfikacji wartości parametru `operation_original_amount` po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe. ---

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`operation_original_currency`	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[12] . Przykład: `operation_original_currency` = EUR --- **Ważne:** Brak weryfikacji wartości parametru `operation_original_currency` po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe. ---
`operation_datetime`	Data realizacji operacji (transakcji) lub zmiany statusu operacji. Format: *YYYY-MM-DD hh:mm:ss* Przykład: `operation_datetime` = 2014-06-01 12:06:37
`operation_related_number`	Numer operacji (transakcji) powiązanej, jeśli takowa istnieje. Format numeru: `^M\d{4,5}\-\d{4,5}$` Przykład: `operation_related_number` = M1234-56789 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.
`control`	Wartość odpowiada parametrowi jaki został przesłany przez serwis sprzedawcy w przekierowaniu do serwisu Dotpay podczas zlecania płatności dla kupującego (parametr `control` w *Tabela 2. Dodatkowe parametry przesyłane do serwisu Dotpay*). typ: *string* maksymalna długość: *1000* Przykład: `control` = ec4bf09d3dbe0cb71e6abc3ea44a7273
`description`	Wartość odpowiada parametrowi jaki został przesłany przez serwis sprzedawcy w przekierowaniu do serwisu Dotpay podczas zlecania płatności dla kupującego (parametr `description` w *Tabela 1. Podstawowe parametry przesyłane do serwisu Dotpay*) typ: *string* minimalna długość: *1* maksymalna długość: *255* Przykład: `description` = Faktura VAT 120/2014

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`email`	Adres e mail podany przez osobę dokonującą płatność. typ: *string* maksymalna długość: *100* Przykład: `email` = jan.nowak@example.com
`p_info`	Nazwa odbiorcy płatności, która została wyświetlona klientowi na stronie płatności serwisu Dotpay podczas dokonywania płatności. typ: *string* maksymalna długość: *300* Przykład: `p_info` = Sklep komputerowy
`p_email`	Adres e-mail, który został wyświetlony kupującemu w celu kontaktu ze sprzedawcą. typ: *string* 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. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_issuer_identification_number` = 603753
`credit_card_masked_number`	Parametr opcjonalny. Zamaskowany numer karty płatniczej, którą wykonana została płatność. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_masked_number` = XXXX XXXX XXXX 6214

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`credit_card_expiration_year`	Parametr opcjonalny. Rok daty ważności karty płatniczej, którą wykonana została płatność. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_expiration_year` = 2019
`credit_card_expiration_month`	Parametr opcjonalny. Miesiąc daty ważności karty płatniczej, którą wykonana została płatność. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_expiration_month` = 01
`credit_card_brand_codename`	Parametr opcjonalny. Nazwa marki karty płatniczej, którą wykonana została płatność. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_brand_codename` = visa
`credit_card_brand_code`	Parametr opcjonalny. Kod marki karty płatniczej, którą wykonana została płatność. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_brand_code` = Visa
`credit_card_unique_identifier`	Parametr opcjonalny. Unikalny identyfikator karty zarejestrowanej w Dotpay. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). --- Przykład: `credit_card_unique_identifier` = 9b73chvjxofy4d9g...d78d7l53ju34po12

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`credit_card_id`	Parametr opcjonalny. Identyfikator karty nadany przez system Dotpay. --- **Ważne:** Do wysłania parametru wymagana jest aktywacja opcji *HTTPS verify* oraz *SSL certificate verify* (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc –> Edycja*). Dodatkowo dane konto (`id`) musi posiadać wymuszoną weryfikację *CHK* po stronie Dotpay. --- Przykład: `credit_card_id` = 59f92e2bf8bedc36...1369dvpa4b7ab02ae
`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[13] (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[14] (alfa-3). Przykład: `geoip_country` = POL

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

PARAMETR	ZNACZENIE / OPIS
`signature`	Suma kontrolna będąca wynikiem działania funkcji skrótu *SHA-256* z konkatenacji powyższych parametrów według poniższego wzoru: *PIN* + `id` + `operation_number` + `operation_type` + `operation_status` + `operation_amount` + `operation_currency` + `operation_withdrawal_amount` + `operation_commission_amount` + `is_completed` + `operation_original_amount` + `operation_original_currency` + `operation_datetime` + `operation_related_number` + `control` + `description` + `email` + `p_info` + `p_email` + `credit_card_issuer_identification_number` + `credit_card_masked_number` + `credit_card_expiration_year` + `credit_card_expiration_month` + `credit_card_brand_codename` + `credit_card_brand_code` + `credit_card_unique_identifier` + `credit_card_id` + `channel` + `channel_country` + `geoip_country` **Ważne:** 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. **Informacja:** *PIN* używany do wyliczania parametru signature to ciąg znaków, który sprzedawca musi wygenerować / określić dla danego sklepu (`id`) w panelu Dotpay po zalogowaniu (menu *Ustawienia –> Powiadomienia –> Konfiguracja urlc*). Służy on jedynie do wyliczenia sumy kontrolnej i **NIE powinien** być wysyłany w danych płatności! **Ważne:** Brak weryfikacji wartości parametru `signature` po stronie systemu sprzedawcy jest niebezpieczne i może narazić na straty finansowe.

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

```php
<?php

$PIN = "Np3n4QmXxp6MOTrLCVs905fdrGf3QIGm";

$sign =
        $PIN.
        $_POST['id'].
        $_POST['operation_number'].
```

(ciąg dalszy na następnej stronie)

[11] https://pl.wikipedia.org/wiki/ISO_4217
[12] https://pl.wikipedia.org/wiki/ISO_4217
[13] https://pl.wikipedia.org/wiki/ISO_3166-1
[14] https://pl.wikipedia.org/wiki/ISO_3166-1

(kontynuacja poprzedniej strony)

```
        $_POST['operation_type'].
        $_POST['operation_status'].
        $_POST['operation_amount'].
        $_POST['operation_currency'].
        $_POST['operation_withdrawal_amount'].
        $_POST['operation_commission_amount'].
        $_POST['is_completed'].
        $_POST['operation_original_amount'].
        $_POST['operation_original_currency'].
        $_POST['operation_datetime'].
        $_POST['operation_related_number'].
        $_POST['control'].
        $_POST['description'].
        $_POST['email'].
        $_POST['p_info'].
        $_POST['p_email'].
        $_POST['credit_card_issuer_identification_number'].
        $_POST['credit_card_masked_number'].
        $_POST['credit_card_expiration_year'].
        $_POST['credit_card_expiration_month'].
        $_POST['credit_card_brand_codename'].
        $_POST['credit_card_brand_code'].
        $_POST['credit_card_unique_identifier'].
        $_POST['credit_card_id'].
        $_POST['channel'].
        $_POST['channel_country'].
        $_POST['geoip_country'];

$signature=hash('sha256', $sign);

?>
```

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). Wartością parametru `chk` jest suma kontrolna będąca wynikiem działania funkcji skrótu SHA-256 z konkatenacji poniższych parametrów z zachowaniem wymienionej kolejności.

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

surcharge + ignore_last_payment_channel + vco_call_id + vco_update_order_info + vco_subtotal + vco_shipping_handling + vco_tax + vco_discount + vco_gift_wrap + vco_misc + vco_promo_code + credit_card_security_code_required + credit_card_operation_type + credit_card_avs + credit_card_threeds + customer + gp_token + blik_refusenopayid + auto_reject_date + ap_token + id1 + amount1 + currency1 + description1 + control1 + … + id(n) + amount(n) + currency(n) + description(n) + control(n)

Ostrzeżenie: Znak + w powyższym wzorze został użyty wyłącznie dla uzyskania czytelności. **NIE** jest wykorzystywany podczas wyliczania sumy kontrolnej.

W przypadku linków wygenerowanych za pomocą panelu administracyjnego (pid), wyliczanie wartości chk odbywa się na zasadzie *PIN* + pid (zgodnie z powyższym wzorem), nie należy uwzględniać parametrów, z których powstał sam pid .

Jeśli w przekierowaniu nie są wykorzystywane dane parametry, należy pominąć je w wyliczeniu.

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).

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 .

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

4.1.1 Przykład 1 podstawowy

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

```php
<?php

################### https://www.dotpay.pl/developer/doc/api_payment/   ############
↪##########################################################
#
#        Exemplary function (PHP) generating  the correct payment redirection␣
↪(POST / GET) to Dotpay payment api with parametr 'chk' (checksum).
#        You enter the payment data in the parameter: $ParametersArray.
#
#
#        Dotpay Sp. z o.o.
#        Tech Customer Service: tech@dotpay.pl
#   Date: 2020-02-11
#
################################################################################
↪############################################################


/** ---------  BASE CONFIG  ---------  **/

// Your Dotpay ID shop (6 digits)
        $DotpayId = "123456";

// PIN for Your Dotpay ID (copy this from your dotpay panel carefully, without␣
↪space)
        $DotpayPin = "MyDotpayPIN000000j4suuSgMPXlg100";

// Dotpay Environment, available: "test" or "production"
        $Environment = "test";

//Redirection method: POST or GET ; recommended method is "POST"
        $RedirectionMethod = "POST";

/**  ---------  end config  ---------  **/



// ** -----------------------   SAMPLE DATA ------------------------- **/

/*  ## SAMPLE PAYMENT DATA IN ##  */
        // Note! You can use more parameters if You need
        // You must give at least: 'amount', 'currency', 'description' (and of␣
↪course ID and PIN in the configuration of this script)
        // see more: https://www.dotpay.pl/developer/doc/api_payment/en/index.html
↪#tabela-1-podstawowe-parametry-przesylane-do-serwisu-dotpay
        // and: https://www.dotpay.pl/developer/doc/api_payment/en/index.html
↪#tabela-2-dodatkowe-parametry-przesylane-do-serwisu-dotpay

// ------
$ParametersArray = array(

        "api_version" => "dev",
        "amount" => "100.00",
        "currency" => "PLN",
        "description" => "Order no. 567915976",
        "url" => "https://www.example.com/thanks_page.php",
        "type" => "0",
        "buttontext" => "Back to the www.example.com",
        "urlc" => "https://www.example.com/urlc_receiver.php",
        "control" => "M1231MzaUdLQWR3",
        "firstname" => "Jan",
        "lastname" => "Nowak",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        "email" => "jan.nowak@example.com",
        "street" => "Warszawska",
        "street_n1" => "1",
        "city" => "Krakow",
        "postcode" => "12-345",
        "phone" => "123456789",
        "country" => "POL",
        "ignore_last_payment_channel" => 1
);

// ** ----------------------   SAMPLE DATA  end ------------------------ **/


## CALCULATE CHECKSUM - CHK

function GenerateChk($DotpayId, $DotpayPin, $ParametersArray)

{
    $ParametersArray['id'] = $DotpayId;

    $CHkInputString =   $DotpayPin.
                        (isset($ParametersArray['api_version']) ? $ParametersArray['api_version'] : null).
                        (isset($ParametersArray['lang']) ? $ParametersArray['lang'] : null).
                        (isset($ParametersArray['id']) ? $ParametersArray['id'] : null).
                        (isset($ParametersArray['amount']) ? $ParametersArray['amount'] : null).
                        (isset($ParametersArray['currency']) ? $ParametersArray['currency'] : null).
                        (isset($ParametersArray['description']) ? $ParametersArray['description'] : null).
                        (isset($ParametersArray['control']) ? $ParametersArray['control'] : null).
                        (isset($ParametersArray['channel']) ? $ParametersArray['channel'] : null).
                        (isset($ParametersArray['url']) ? $ParametersArray['url'] : null).
                        (isset($ParametersArray['type']) ? $ParametersArray['type'] : null).
                        (isset($ParametersArray['buttontext']) ? $ParametersArray['buttontext'] : null).
                        (isset($ParametersArray['urlc']) ? $ParametersArray['urlc'] : null).
                        (isset($ParametersArray['firstname']) ? $ParametersArray['firstname'] : null).
                        (isset($ParametersArray['lastname']) ? $ParametersArray['lastname'] : null).
                        (isset($ParametersArray['email']) ? $ParametersArray['email'] : null).
                        (isset($ParametersArray['street']) ? $ParametersArray['street'] : null).
                        (isset($ParametersArray['street_n1']) ? $ParametersArray['street_n1'] : null).
                        (isset($ParametersArray['city']) ? $ParametersArray['city'] : null).
                        (isset($ParametersArray['postcode']) ? $ParametersArray['postcode'] : null).
                        (isset($ParametersArray['phone']) ? $ParametersArray['phone'] : null).
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
                        (isset($ParametersArray['country']) ? $ParametersArray[
↪'country'] : null).
                        (isset($ParametersArray['ignore_last_payment_channel']) ?
↪$ParametersArray['ignore_last_payment_channel'] : null);

    return hash('sha256',$CHkInputString);
}



## GENERATE FORM TO DOTPAY

function GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
↪$RedirectionMethod, $ParametersArray)
{
    $ChkValue = GenerateChk($DotpayId, $DotpayPin, $ParametersArray);

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

    if ($RedirectionMethod == 'POST') {
        $RedirectionCode = '<form action="'.$EnvironmentAddress.'" method="POST"␣
↪id="dotpay_redirection_form" accept-charset="UTF-8">'.PHP_EOL;
                $RedirectionCode .= "\t".'<input name="id" value="'.$DotpayId.'"␣
↪type="hidden"/>'.PHP_EOL;

        foreach ($ParametersArray as $key => $value)
                {
            $RedirectionCode .= "\t".'<input name="'.$key.'" value="'.$value.'"␣
↪type="hidden"/>'.PHP_EOL;
        }
        $RedirectionCode .= "\t".'<input name="chk" value="'.$ChkValue.'" type=
↪"hidden"/>'.PHP_EOL;
        $RedirectionCode .= '</form>'.PHP_EOL.'<button id="dotpay_redirection_
↪button" type="submit" form="dotpay_redirection_form" value="Submit">Confirm and␣
↪Pay</button>'.PHP_EOL;

        return $RedirectionCode;

    } elseif ($RedirectionMethod == 'GET') {
        $RedirectionCode = $EnvironmentAddress.'?';

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

        $RedirectionCode .= 'id='.$DotpayId;
        $RedirectionCode .= '&chk='.$ChkValue;

                return '<a href="'.$RedirectionCode.'">Link to Pay</a>';
    }
}


##  get form (POST method) or payment link (GET method)
##  ("Dotpay ID","PIN","[test|production]","[POST|GET]","payment data")

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
150 echo GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
    ↪$RedirectionMethod , $ParametersArray);
151
152 ?>
```

4.1.2 Przykład 2 pełny

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  #        Exemplary function (PHP) generating  the correct payment redirection␣
   ↪(POST / GET) to Dotpay payment api with parametr 'chk' (checksum).
6  #        You enter the payment data in the parameter: $ParametersArray.
7  #
8  #        In addition, the examplary includes the use of data for the Multimerchant␣
   ↪service ($MultiMerchantList)
9  #        and some payment channels that require additional data, e.g. a delivery␣
   ↪address ($customer).
10 #        You do not need to use them if you do not use these features.
11 #
12 #        Dotpay Sp. z o.o.
13 #        Tech Customer Service: tech@dotpay.pl
14 #   Date: 2019-04-15
15 #
16 ##########################################################################################
   ↪#############################################################
17
18
19 /** ---------  BASE CONFIG  ---------  **/
20
21 // Your Dotpay ID shop (6 digits)
22         $DotpayId = "123456";
23
24 // PIN for Your Dotpay ID (copy this from your dotpay panel carefully, without␣
   ↪space)
25         $DotpayPin = "Oi9xSl8lE4cJhygjKn2L1MW2MBqkv234";
26
27 // Dotpay Environment, available: "test" or "production"
28         $Environment = "test";
29
30 //Redirection method: POST or GET ; recommended method is "POST"
31         $RedirectionMethod = "POST";
32
33 /**  ---------  end config  ---------  **/
34
35
36 // Do not remove this!
37 $MultiMerchantList = array(); //optional data
38 $customer = null;  //optional data
39
40
41 ## CALCULATE CHECKSUM - CHK
42
43 function GenerateChk($DotpayId, $DotpayPin, $Environment, $RedirectionMethod,
   ↪$ParametersArray, $MultiMerchantList, $customer_base64)
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
44
45  {
46      $ParametersArray['id'] = $DotpayId;
47          $ParametersArray['customer'] = $customer_base64;
48
49      $chk =   $DotpayPin.
50      (isset($ParametersArray['api_version']) ? $ParametersArray['api_version'] :␣
    ↪null).
51      (isset($ParametersArray['lang']) ? $ParametersArray['lang'] : null).
52      (isset($ParametersArray['id']) ? $ParametersArray['id'] : null).
53      (isset($ParametersArray['pid']) ? $ParametersArray['pid'] : null).
54      (isset($ParametersArray['amount']) ? $ParametersArray['amount'] : null).
55      (isset($ParametersArray['currency']) ? $ParametersArray['currency'] : null).
56      (isset($ParametersArray['description']) ? $ParametersArray['description'] :␣
    ↪null).
57      (isset($ParametersArray['control']) ? $ParametersArray['control'] : null).
58      (isset($ParametersArray['channel']) ? $ParametersArray['channel'] : null).
59      (isset($ParametersArray['credit_card_brand']) ? $ParametersArray['credit_card_
    ↪brand'] : null).
60      (isset($ParametersArray['ch_lock']) ? $ParametersArray['ch_lock'] : null).
61      (isset($ParametersArray['channel_groups']) ? $ParametersArray['channel_groups
    ↪'] : null).
62      (isset($ParametersArray['onlinetransfer']) ? $ParametersArray['onlinetransfer
    ↪'] : null).
63      (isset($ParametersArray['url']) ? $ParametersArray['url'] : null).
64      (isset($ParametersArray['type']) ? $ParametersArray['type'] : null).
65      (isset($ParametersArray['buttontext']) ? $ParametersArray['buttontext'] :␣
    ↪null).
66      (isset($ParametersArray['urlc']) ? $ParametersArray['urlc'] : null).
67      (isset($ParametersArray['firstname']) ? $ParametersArray['firstname'] : null).
68      (isset($ParametersArray['lastname']) ? $ParametersArray['lastname'] : null).
69      (isset($ParametersArray['email']) ? $ParametersArray['email'] : null).
70      (isset($ParametersArray['street']) ? $ParametersArray['street'] : null).
71      (isset($ParametersArray['street_n1']) ? $ParametersArray['street_n1'] : null).
72      (isset($ParametersArray['street_n2']) ? $ParametersArray['street_n2'] : null).
73      (isset($ParametersArray['state']) ? $ParametersArray['state'] : null).
74      (isset($ParametersArray['addr3']) ? $ParametersArray['addr3'] : null).
75      (isset($ParametersArray['city']) ? $ParametersArray['city'] : null).
76      (isset($ParametersArray['postcode']) ? $ParametersArray['postcode'] : null).
77      (isset($ParametersArray['phone']) ? $ParametersArray['phone'] : null).
78      (isset($ParametersArray['country']) ? $ParametersArray['country'] : null).
79      (isset($ParametersArray['code']) ? $ParametersArray['code'] : null).
80      (isset($ParametersArray['p_info']) ? $ParametersArray['p_info'] : null).
81      (isset($ParametersArray['p_email']) ? $ParametersArray['p_email'] : null).
82      (isset($ParametersArray['n_email']) ? $ParametersArray['n_email'] : null).
83      (isset($ParametersArray['expiration_date']) ? $ParametersArray['expiration_date
    ↪'] : null).
84      (isset($ParametersArray['deladdr']) ? $ParametersArray['deladdr'] : null).
85      (isset($ParametersArray['recipient_account_number']) ? $ParametersArray[
    ↪'recipient_account_number'] : null).
86      (isset($ParametersArray['recipient_company']) ? $ParametersArray['recipient_
    ↪company'] : null).
87      (isset($ParametersArray['recipient_first_name']) ? $ParametersArray['recipient_
    ↪first_name'] : null).
88      (isset($ParametersArray['recipient_last_name']) ? $ParametersArray['recipient_
    ↪last_name'] : null).
89      (isset($ParametersArray['recipient_address_street']) ? $ParametersArray[
    ↪'recipient_address_street'] : null).
90      (isset($ParametersArray['recipient_address_building']) ? $ParametersArray[
    ↪'recipient_address_building'] : null).
91      (isset($ParametersArray['recipient_address_apartment']) ? $ParametersArray[
    ↪'recipient_address_apartment'] : null).
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
92      (isset($ParametersArray['recipient_address_postcode']) ? $ParametersArray[
↪'recipient_address_postcode'] : null).
93      (isset($ParametersArray['recipient_address_city']) ? $ParametersArray[
↪'recipient_address_city'] : null).
94      (isset($ParametersArray['application']) ? $ParametersArray['application'] :␣
↪null).
95      (isset($ParametersArray['application_version']) ? $ParametersArray[
↪'application_version'] : null).
96      (isset($ParametersArray['warranty']) ? $ParametersArray['warranty'] : null).
97      (isset($ParametersArray['bylaw']) ? $ParametersArray['bylaw'] : null).
98      (isset($ParametersArray['personal_data']) ? $ParametersArray['personal_data']␣
↪: null).
99      (isset($ParametersArray['credit_card_number']) ? $ParametersArray['credit_card_
↪number'] : null).
100     (isset($ParametersArray['credit_card_expiration_date_year']) ?
↪$ParametersArray['credit_card_expiration_date_year'] : null).
101     (isset($ParametersArray['credit_card_expiration_date_month']) ?
↪$ParametersArray['credit_card_expiration_date_month'] : null).
102     (isset($ParametersArray['credit_card_security_code']) ? $ParametersArray[
↪'credit_card_security_code'] : null).
103     (isset($ParametersArray['credit_card_store']) ? $ParametersArray['credit_card_
↪store'] : null).
104     (isset($ParametersArray['credit_card_store_security_code']) ? $ParametersArray[
↪'credit_card_store_security_code'] : null).
105     (isset($ParametersArray['credit_card_customer_id']) ? $ParametersArray['credit_
↪card_customer_id'] : null).
106     (isset($ParametersArray['credit_card_id']) ? $ParametersArray['credit_card_id
↪'] : null).
107     (isset($ParametersArray['blik_code']) ? $ParametersArray['blik_code'] : null).
108     (isset($ParametersArray['credit_card_registration']) ? $ParametersArray[
↪'credit_card_registration'] : null).
109     (isset($ParametersArray['surcharge_amount']) ? $ParametersArray['surcharge_
↪amount'] : null).
110     (isset($ParametersArray['surcharge']) ? $ParametersArray['surcharge'] : null).
111     (isset($ParametersArray['surcharge']) ? $ParametersArray['surcharge'] : null).
112     (isset($ParametersArray['ignore_last_payment_channel']) ? $ParametersArray[
↪'ignore_last_payment_channel'] : null).
113     (isset($ParametersArray['vco_call_id']) ? $ParametersArray['vco_call_id'] :␣
↪null).
114     (isset($ParametersArray['vco_update_order_info']) ? $ParametersArray['vco_
↪update_order_info'] : null).
115     (isset($ParametersArray['vco_subtotal']) ? $ParametersArray['vco_subtotal'] :␣
↪null).
116     (isset($ParametersArray['vco_shipping_handling']) ? $ParametersArray['vco_
↪shipping_handling'] : null).
117     (isset($ParametersArray['vco_tax']) ? $ParametersArray['vco_tax'] : null).
118     (isset($ParametersArray['vco_discount']) ? $ParametersArray['vco_discount'] :␣
↪null).
119     (isset($ParametersArray['vco_gift_wrap']) ? $ParametersArray['vco_gift_wrap']␣
↪: null).
120     (isset($ParametersArray['vco_misc']) ? $ParametersArray['vco_misc'] : null).
121     (isset($ParametersArray['vco_promo_code']) ? $ParametersArray['vco_promo_code
↪'] : null).
122     (isset($ParametersArray['credit_card_security_code_required']) ?
↪$ParametersArray['credit_card_security_code_required'] : null).
123     (isset($ParametersArray['credit_card_operation_type']) ? $ParametersArray[
↪'credit_card_operation_type'] : null).
124     (isset($ParametersArray['credit_card_avs']) ? $ParametersArray['credit_card_avs
↪'] : null).
125     (isset($ParametersArray['credit_card_threeds']) ? $ParametersArray['credit_
↪card_threeds'] : null).
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
126     (isset($ParametersArray['customer']) ? $ParametersArray['customer'] : null).
127     (isset($ParametersArray['gp_token']) ? $ParametersArray['gp_token'] : null).
128     (isset($ParametersArray['blik_refusenopayid']) ? $ParametersArray['blik_
    ↪refusenopayid'] : null).
129     (isset($ParametersArray['auto_reject_date']) ? $ParametersArray['auto_reject_
    ↪date'] : null).
130     (isset($ParametersArray['ap_token']) ? $ParametersArray['ap_token'] : null);
131 
132     foreach ($MultiMerchantList as $item) {
133         foreach ($item as $key => $value) {
134             $chk =   $chk.
135 (isset($value) ? $value : null);
136         }
137     }
138     return $chk;
139 }
140 
141 
142 
143 ## GENERATE FORM TO DOTPAY
144 
145 function GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment,
    ↪$RedirectionMethod, $ParametersArray, $MultiMerchantList, $customer_base64)
146 {
147     $ParametersArray['id'] = $DotpayId;
148     $ChkParametersChain = GenerateChk($DotpayId, $DotpayPin, $Environment,
    ↪$RedirectionMethod, $ParametersArray, $MultiMerchantList, $customer_base64);
149 
150 
151     $ChkValue = hash('sha256', $ChkParametersChain);
152 
153     if ($Environment == 'production') {
154         $EnvironmentAddress = 'https://ssl.dotpay.pl/t2/';
155     } elseif ($Environment == 'test') {
156         $EnvironmentAddress = 'https://ssl.dotpay.pl/test_payment/';
157     }
158 
159     if ($RedirectionMethod == 'POST') {
160         $RedirectionCode = '<form action="'.$EnvironmentAddress.'" method="POST"␣
    ↪id="dotpay_redirection_form" accept-charset="UTF-8">'.PHP_EOL;
161 
162         foreach ($ParametersArray as $key => $value) {
163             $RedirectionCode .= "\t".'<input name="'.$key.'" value="'.$value.'"␣
    ↪type="hidden"/>'.PHP_EOL;
164         }
165 
166                 if(isset($customer_base64)) {
167                         $RedirectionCode .= "\t".'<input name="customer" value="'.
    ↪$customer_base64.'" type="hidden"/>'.PHP_EOL;
168                 }
169 
170         foreach ($MultiMerchantList as $item) {
171             foreach ($item as $key => $value) {
172                 $RedirectionCode .= "\t".'<input name="'.$key.'" value="'.$value.'
    ↪" type="hidden"/>'.PHP_EOL;
173             }
174         }
175 
176         $RedirectionCode .= "\t".'<input name="chk" value="'.$ChkValue.'" type=
    ↪"hidden"/>'.PHP_EOL;
177         $RedirectionCode .= '</form>'.PHP_EOL.'<button id="dotpay_redirection_
    ↪button" type="submit" form="dotpay_redirection_form" value="Submit">Confirm and␣
    ↪Pay</button>'.PHP_EOL;
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
178
179         return $RedirectionCode;
180
181     } elseif ($RedirectionMethod == 'GET') {
182         $RedirectionCode = $EnvironmentAddress.'?';
183
184         foreach ($ParametersArray as $key => $value) {
185             $RedirectionCode .= $key.'='.rawurlencode($value).'&';
186         }
187
188                 if(isset($customer_base64)) {
189                         $RedirectionCode .= 'customer='.$customer_base64.'&';
190                 }
191
192         foreach ($MultiMerchantList as $item) {
193             foreach ($item as $key => $value) {
194                 $RedirectionCode .= $key.'='.rawurlencode($value).'&';
195             }
196         }
197
198         $RedirectionCode .= 'chk='.$ChkValue;
199
200                 return '<a href="'.$RedirectionCode.'">Go to Pay</a><br>link:<br>'.
    ↪$RedirectionCode;
201     }
202 }
203
204
205
206
207
208 // ** ----------------------   SAMPLE DATA ------------------------- **/
209
210
211
212 /*  ## SAMPLE PAYMENT DATA IN ##  */
213         // Note! You can use more parameters if You need
214         // You must give at least: 'amount', 'currency', 'description' (and of␣
    ↪course ID and PIN in the configuration of this script)
215         // see more: https://www.dotpay.pl/developer/doc/api_payment/en/index.html
    ↪#tabela-1-podstawowe-parametry-przesylane-do-serwisu-dotpay
216         // and: https://www.dotpay.pl/developer/doc/api_payment/en/index.html
    ↪#tabela-2-dodatkowe-parametry-przesylane-do-serwisu-dotpay
217
218 // ------
219 $ParametersArray = array(
220         "api_version" => "dev",
221         "amount" => "100.00",
222         "currency" => "PLN",
223         "description" => "Order no. 567915976",
224         "url" => "https://www.example.com/thanks_page.php",
225         "type" => "4",
226         "channel" => "2",
227         "buttontext" => "Back to the www.example.com",
228         "urlc" => "https://www.example.com/urlc_receiver.php",
229         "control" => "M1231MzaUdLQWR3",
230         "firstname" => "Jan",
231         "lastname" => "Nowak",
232         "email" => "jan.nowak@example.com",
233         "street" => "Warszawska",
234         "street_n1" => "1",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        "city" => "Krakow",
        "postcode" => "12-345",
        "phone" => "123456789",
        "country" => "POL",
        "ignore_last_payment_channel" => "true"
);
// ------

/*   ###  SAMPLE CUSTOMER DATA IN with delivery address (optional) ###
                        You can remove it if You don't need it
*/

// ------
$customer = array(
        "payer" => array(
                "first_name" => "Jan",
                "last_name" => "Kowal",
                "email" => "mymail@example.com",
        ) ,
        "order" => array(
                "delivery_address" => array(
                        "city" => "Kraków",
                        "street" => "Wielicka",
                        "building_number" => "28B",
                        "postcode" => "30-552",
                )
        )
);

// ------



/*  ### SAMPLE Multimerchant DATA IN (3 accounts 'child' type ) - optional  ###
                 You can remove it if You don't need it
                 see more: https://www.dotpay.pl/developer/doc/api_payment/en/
 ↪index.html#platnosc-dzielona-multimerchant-pasaz
*/

// ------ uncomment if you need:

/*
        $MultiMerchantList = array(
                $MultiMerchant1 = array(
                        "id1" => "123456",
                        "amount1" => "10.00",
                        "currency1" => "PLN",
                        "description1" => "description1",
                        "control1" => "control1",
                ) ,
                $MultiMerchant2 = array(
                        "id2" => "234561",
                        "amount2" => "60.00",
                        "currency2" => "PLN",
                        "description2" => "description2",
                        "control2" => "control2",
                ) ,
                $MultiMerchant3 = array(
                        "id3" => "234562",
                        "amount3" => "30.00",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
                        "currency3" => "PLN",
                        "description3" => "description3",
                        "control3" => "control3",
                )
        );

*/
// ------

// ** ----------------------   SAMPLE DATA  end ------------------------ **/


if (empty($customer) || !isset($customer['payer']) || !isset($customer['order']['delivery_address'])) {
        $customer_base64 = null;
} else {
        $customer_base64 = base64_encode(json_encode($customer));
}


##  get form (POST method) or payment link (GET method)
##  ("Dotpay ID","PIN","[test|production]","[POST|GET]","payment data","Multimerchant data", "additional customer data")

echo GenerateChkDotpayRedirection($DotpayId, $DotpayPin, $Environment, $RedirectionMethod , $ParametersArray, $MultiMerchantList, $customer_base64);


?>
```

4.2 One-click i płatności cykliczne

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

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`	Unikalny identyfikator płacącego nadany przez system sprzedawcy opisany w powyższej tabeli.

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 *URLC* (wymagana wcześniejsza konfiguracja konta, opis w rozdz. *Odbieranie informacji po płatności (powiadomienia urlc)*, alternatywnie może być pobrany ze szczegółów wykonanej operacji płatności za pośrednictwem API panelu administracyjnego sprzedawcy[15]

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)

PARAMETR	ZNACZENIE / OPIS
`credit_card_operation_type`	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. --- **Wskazówka:** Karta zarejestrowana poprzez e_commerce nie może zostać wykorzystana do wykonania operacji typu recurring. Natomiast karta rejestrowana poprzez recurring_init może zostać wykorzystana zarówno do płatności typu recurring jak i e_commerce. ---

ciąg dalszy na następnej stronie

[15] https://www.dotpay.pl/developer/doc/api_seller/pl/#szczegoly-pojedynczej-operacji

Table 6 – kontynuacja poprzedniej strony

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

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:

```
<div>

  <form action="https://ssl.dotpay.pl/t2/" method="post" id="dotpay_redirection_
↪form">
    <input name="api_version" value="dev" type="hidden" />
    <input name="id" value="123456" type="hidden" />
    <input name="amount" value="320.00" type="hidden" />
    <input name="currency" value="PLN" type="hidden" />
    <input name="description" value="Płatność za 12345/2014" type="hidden" />
    <input name="control" value="202cb962ac590" type="hidden" />
    <input name="channel" value="248" type="hidden" />
    <input name="ch_lock" value="1" type="hidden" />
    <input name="firstname" value="John" type="hidden" />
    <input name="lastname" value="Smith" type="hidden" />
    <input name="email" value="john.smith@example.com" type="hidden" />
    <input name="type" value="0" type="hidden" />
    <input name="url" value="https://www.example.com/thanks_page.php" type="hidden
↪" />
    <input name="urlc" value="https://www.example.com/urlc_receiver.php" type=
↪"hidden" />
    <input name="credit_card_store" value="1" type="hidden" />
    <input name="credit_card_customer_id" value="f9c6a4-25473" type="hidden" />
    <input name="chk" value="11ac1938ac47ddd53815b4aeb6230ab9fe4554d82ee11e39c41b9055f38f5c08
↪" type="hidden" />
  </form>
  <p>
    <button type="submit" form="dotpay_redirection_form" value="Submit">
      Potwierdź zamówienie i zapłać</button>
  </p>

</div>
```

Przykładowe żądanie płatności z wykorzystaniem zarejestrowanych danych karty klienta (one-click):

```
<div>
  <form action="https://ssl.dotpay.pl/t2/" method="post" id="dotpay_redirection_
↪form">

    <input name="api_version" value="dev" type="hidden" />
    <input name="id" value="123456" type="hidden" />
    <input name="amount" value="410.00" type="hidden" />
    <input name="currency" value="PLN" type="hidden" />
    <input name="description" value="Płatność za 12346/2014" type="hidden" />
    <input name="control" value="31ee79b30dc39a9cbaa" type="hidden" />
    <input name="channel" value="248" type="hidden" />
    <input name="ch_lock" value="1" type="hidden" />
    <input name="firstname" value="John" type="hidden" />
    <input name="lastname" value="Smith" type="hidden" />
    <input name="email" value="john.smith@example.com" type="hidden" />
    <input name="type" value="4" type="hidden" />
    <input name="url" value="https://www.example.com/thanks_page.php" type="hidden
↪" />
    <input name="urlc" value="https://www.example.com/urlc_receiver.php" type=
↪"hidden" />
    <input name="credit_card_customer_id" value="f9c6a4-25473" type="hidden" />
    <input name="credit_card_id" value="59f92e2bf8bedc36bec2219862448dd54d...
↪1829a239eb7432d0easuxp2w158eb13d6333ce71369184eb7ab02ae" type="hidden" />
    <input name="chk" value="ed0ef4e488ec2de3135b0a1ca226c31867f78bbcd8fe06506ae666210a78d68c
↪" type="hidden" />
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
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 bezpośredniej 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).

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: *credit_card_number* maksymalna długość: *26* wyrażenie regularne: `^[\d\s]{12,26}$` Przykład: `credit_card_number` = 5500005555555559
`credit_card_expiration_date_year`	Rok daty ważności karty płatniczej klienta. długość: *4* wyrażenie regularne: `^(20)((19)|([2-9][0-9]))$` Przykład: `credit_card_expiration_date_year` = 2019
`credit_card_expiration_date_month`	Miesiąc daty ważności karty płatniczej klienta. długość: *2* wyrażenie regularne: `^(0[1-9])|(1[0-2])$` 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: *number* maksymalna długość: *4* wyrażenie regularne: `^\d{3,4}$` 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:

```php
<?php

$ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, "https://ssl.dotpay.pl/t2/payment_api/v1/cards/59f92e2bf8bedc36bec221...718c58eb13d6333ce71369184eb7ab02ae/");
  curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
  curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
  curl_setopt($ch, CURLOPT_CAINFO, "ca-bundle.crt"); //http://curl.haxx.se/docs/caextract.html
  curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  curl_setopt($ch, CURLOPT_TIMEOUT, 100);
  curl_setopt($ch, CURLOPT_USERPWD, 'user:password');
  curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");

  $response = curl_exec($ch); // API response
  $curl_info = curl_getinfo($ch); //curl info
  curl_close($ch);

  echo '<pre>';
  echo 'HTTP status code: '.$curl_info[http_code];
  echo PHP_EOL.'-------------------------'.PHP_EOL.PHP_EOL;
    print_r(json_decode($response));
    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[16]	Usunięto
404 Not Found[17]	Nie znaleziono karty
400 Bad Request[18]	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 zależne 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.) przesłanych 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)

[16] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.5
[17] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5
[18] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1

(kontynuacja poprzedniej strony)

```
  <form action="https://ssl.dotpay.pl/t2/" id="dotpay_redirection_form" method="POST" enctype="application/x-www-form-urlencoded">

    <input type="hidden" name="id" value="123456">
    <input type="hidden" name="amount" value="320.00">
    <input type="hidden" name="currency" value="PLN">
    <input type="hidden" name="description" value="Płatność za zamówienie 01/2017␣parent">
    <input type="hidden" name="control" value="control_parent">

    <input type="hidden" name="id1" value="456123">
    <input type="hidden" name="amount1" value="120.00">
    <input type="hidden" name="currency1" value="PLN">
    <input type="hidden" name="description1" value="Płatność za zamówienie 01/2017␣child1">
    <input type="hidden" name="control1" value="control_child1">

    <input type="hidden" name="id2" value="561423">
    <input type="hidden" name="amount2" value="90.00">
    <input type="hidden" name="currency2" value="PLN">
    <input type="hidden" name="description2" value="Płatność za zamówienie 01/2017␣child2">
    <input type="hidden" name="control2" value="control_child2">

    <input type="hidden" name="id3" value="642513">
    <input type="hidden" name="amount3" value="110.00">
    <input type="hidden" name="currency3" value="PLN">
    <input type="hidden" name="description3" value="Płatność za zamówienie 01/2017␣child3">
    <input type="hidden" name="control3" value="control_child3">
  </form>

  <p>
    <button type="submit" form="dotpay_redirection_form" value="Submit">
      Potwierdź zamówienie i zapłać
    </button>
  </p>

</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 osobna 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 dodatkow wykorzystywane w usłudze Masscollect)

PARAMETR	ZNACZENIE / OPIS
`recipient_account_number`	Numer rachunku odbiorcy w formacie BBAN[19] . typ: *string* maksymalna długość: *26* Przykład: `recipient_account_number` = 32249000896640389235035459
`recipient_company`	Nazwa firmy odbiorcy płatności typ: *string* maksymalna długość: *50* Przykład: `recipient_company` = Moja Firma S.A.
`recipient_first_name`	Imię odbiorcy płatności typ: *string* maksymalna długość: *30* Przykład: `recipient_first_name` = Jan
`recipient_last_name`	Nazwisko odbiorcy płatności typ: *string* maksymalna długość: *30* Przykład: `recipient_last_name` = Kowalski
`recipient_address_street`	Ulica – adres odbiorcy płatności typ: *string* maksymalna długość: *40* Przykład: `recipient_address_street` = Wielicka
`recipient_address_building`	Numer budynku – adres odbiorcy płatności typ: *string* maksymalna długość: *10* Przykład: `recipient_address_building` = 72
`recipient_address_apartment`	Numer lokalu – adres odbiorcy płatności typ: *string* 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: *string* maksymalna długość: *6* Przykład: `recipient_address_postcode` = 30-552
`recipient_address_city`	Miasto – adres odbiorcy płatności typ: *string* maksymalna długość: *50* Przykład: `recipient_address_city` = Kraków

Poniżej został zamieszczony przykładowy formularz zlecenia płatności w modelu Masscollect.

```
<div>

  <form action="https://ssl.dotpay.pl/t2/" id="dotpay_redirection_form" method="POST" enctype="application/x-www-form-urlencoded">

    <input type="hidden" name="id" value="123456">
    <input type="hidden" name="amount" value="123.45">
    <input type="hidden" name="description" value="Płatność za zamówienie 07/2017">
    <input type="hidden" name="recipient_account_number" value="32249000896640389235035459">
    <input type="hidden" name="recipient_company" value="Moja Firma S.A.">
    <input type="hidden" name="recipient_first_name" value="Jan">
    <input type="hidden" name="recipient_last_name" value="Kowalski">
    <input type="hidden" name="recipient_address_street" value="Wielicka">
    <input type="hidden" name="recipient_address_building" value="72">
    <input type="hidden" name="recipient_address_apartment" value="1">
    <input type="hidden" name="recipient_address_postcode" value="30-552">
    <input type="hidden" name="recipient_address_city" value="Kraków">
    <input type="hidden" name="chk" value="3135b6debcd8fe4e488ec2easux506c31867f78bed0ef0a1ca2266" />
  </form>

  <p>
    <button type="submit" form="dotpay_redirection_form" value="Submit">
      Potwierdź zamówienie i zapłać
    </button>
  </p>

</div>
```

4.5 Obsługa danych dostawy oraz płacącego

W celu poprawnej oceny zdolności kredytowej płacącego 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.

[19] https://pl.wikipedia.org/wiki/Mi%C4%99dzynarodowy_numer_rachunku_bankowego

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 1. Niektóre kanały płatności mogą wymagać mniejszej ilosci danych, jak np. *PayPo*.

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

NAZWA POLA	TYP	OPIS
`payer.first_name`	string	Imię płacącego
`payer.last_name`	string	Nazwisko płacącego
`payer.email`	string	Adres email płacącego
`payer.phone`	string	Numer telefonu płacącego
`payer.address`	-	Adres płacącego / rozliczeniowy
`payer.address.city`	string	Adres płacącego: miasto
`address.street`	string	Adres płacącego: ulica
`address.building_number`	string	Adres płacącego: numer budynku
`address.flat_number`	string	Adres płacącego: numer mieszkania
`address.postcode`	string	Adres płacącego: kod pocztowy
`address.country`	string	Adres płacącego: dwuliterowy (ISO 3166-1 alpha2[20]) lub trzyliterowy (ISO 3166-1 alpha3[21]) kod kraju
`is_logged_in`	boolean	Informacja czy płacący przed dokonaniem płatności był zarejestrowany w systemie sprzedawcy
`registered_since`	string	Data rejestracji płacącego w serwisie sprzedawcy, format *YYYY-MM-DD* lub *YYYY-MM-DD hh:mm:ss* Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: `order_count`
`registered_since_indicator`	string (*indykator*)	Data rejestracji płacącego w serwisie sprzedawcy, indykator dla pola `registered_since` Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: `order_count`
`account_update`	string	Data ostatniej zmiany konta płacącego w serwisie sprzedawcy, format *YYYY-MM-DD*
`account_update_indicator`	string (*indykator*)	Data ostatniej zmiany konta płacącego w serwisie sprzedawcy, indykator dla pola `account_update`
`password_change`	string	Data ostatniej zmiany hasła dla konta płacącego w serwisie sprzedawcy, format *YYYY-MM-DD*
`password_change_indicator`	string (*indykator*)	Data ostatniej zmiany hasła dla konta płacącego w serwisie sprzedawcy, indykator dla pola `password_change`
`shipping_address_since`	string	Data od kiedy podany adres dostawy płacącego w serwisie sprzedawcy jest używany, format *YYYY-MM-DD*
`shipping_address_since_indicator`	string (*indykator*)	Data od kiedy podany adres dostawy płacącego w serwisie sprzedawcy jest używany, indykator dla pola `shipping_address_since`
`order_count`	int	Ilość złożonych zamówień przez płacącego 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 płacącego w serwisie sprzedawcy w tym samym dniu
`order_count_year`	int	Ilość złożonych zamówień przez płacącego 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 płacącego w sklepie

ciąg dalszy na następnej stronie

Table 14 – kontynuacja poprzedniej strony

NAZWA POLA	TYP	OPIS
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
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.
order.items.quantity	int	Ilość pozycji w zamówieniu klienta, czyli: n * Towar = łączna ilość pozycji (minimalnie 1)
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.delivery_type	string	Metoda dostawy Dostępne wartości: • COURIER - kurier • POCZTA_POLSKA - Poczta Polska • PICKUP_POINT - dostawa do punktu (np. UPS Access point, DHL Parcel Shop) • PACZKOMAT - paczkomat • PACZKA_W_RUCHU - paczka w ruchu • PICKUP_SHOP - odbiór w sklepie (click&collect)
order.delivery_address	-	Adres dostawy
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[22]) lub trzyliterowy (ISO 3166-1 alpha3[23]) kod kraju

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

[20] https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-2
[21] https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-3
[22] https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-2
[23] https://pl.wikipedia.org/wiki/ISO_3166-1_alfa-3

4.5.2 Wartości używane dla pola zastępczego typu indykator 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 1: Przykład użycia parametrów (format json)

```
{
    "payer": {
        "first_name": "Jan",
        "last_name": "Kowal",
        "email": "jankowal@example.com",
        "phone": "123456789",
        "address": {
            "city": "Warszawa",
            "street": "Krucza",
            "building_number": "1a",
            "flat_number": "4",
            "postcode": "00-950",
            "country": "POL"
        }
    },
    "is_logged_in": true,
    "registered_since": "2017-02-11",
    "order_count": 2,
    "order_history": [
        {
            "date": "2017-02-11",
            "amount": "456.21",
            "delivery_type": "COURIER"
        },
        {
            "date": "2018-05-21",
            "amount": "879.67",
            "delivery_type": "POCZTA_POLSKA"
        }
    ],
    "order": {
        "id": "MHH67HF8DS",
        "items": [
            {
                "id": "3245623",
                "name": "Super Phone 1",
                "quantity": 1,
                "unit_type": "szt.",
                "gross_price": "856.52",
                "type": "towar",
                "is_virtual": false
            },
            {
                "id": "3245625",
                "name": "Dostawa",
                "quantity": 1,
                "unit_type": "szt.",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
                "gross_price": "15.00",
                "type": "dostawa",
                "is_virtual": false
            }
        ],
        "delivery_type": "POCZTA_POLSKA",
        "delivery_address": {
            "city": "Kraków",
            "street": "Wielicka",
            "building_number": "28b",
            "flat_number": "5",
            "postcode": "30-552",
            "country": "POL"
        }
    }
}
```

4.5.3 Wymagane dane przesyłane w parametrze `customer` dla kanału PayPo

W celu realizacji płatności z wykorzystaniem kanału PayPo, neleży wysłać komplet danych wymaganych przez ten kanał płatności. Wybrane dane z *Tabeli 1* niezbędne do realizacji płatności tym kanałem:

Listing 2: Przykład użycia minimalnej ilości wymaganych danych dla kanału PayPo (format json):

```
    {
      "payer": {
        "first_name": "Jan",
        "last_name": "Kowal",
        "email": "jankowal@example.com"
      },
      "order": {
        "delivery_address": {
          "city": "Kraków",
          "street": "Wielicka",
          "building_number": "28B",
          "postcode": "30-552"
        }
      }
    }
```

Listing 3: Przykład użycia kompletnych danych dla kanału **PayPo** (format json)

```
    {
      "payer": {
        "first_name": "Jan",
        "last_name": "Kowal",
        "email": "jankowal@example.com",
        "phone": "+48126882600"
      },
      "registered_since": "2017-02-11",
      "order_count": 2,
      "order": {
        "id": "MHH67HF8DS",
        "delivery_type": "POCZTA_POLSKA",
        "delivery_address": {
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
            "city": "Kraków",
            "street": "Wielicka",
            "building_number": "28b",
            "flat_number": "5",
            "postcode": "30-552",
            "country": "POL"
        }
      }
    }
```

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

Aby takie dane mogły być przesłane do Dotpay w parametrze `customer` , należy je odpowiednio wcześniej sformatować. Dane te powinny zostać zakodowane do formatu JSON[24] a następnie zakodowane przy użyciu Base64[25].

Listing 4: Przykład przygotowania parametru `customer` dla kanału **PayPo** (wersja minimum) w języku PHP:

```php
<?php

        $customer = array (

            "registered_since" => "2017-12-31",
            "order_count" => 12,


            "payer" => array(
                    "first_name" => "Jan",
                    "last_name" => "Kowal",
                    "email" => "jan@example.com"
                     ),
            "order" => array(
                    "delivery_type" => "COURIER",
                    "delivery_address" => array(

                                    "city" => "Krakow",
                                    "street" => "Wielicka",
                                    "building_number" => "11",
                                    "flat_number" => "7",
                                    "postcode" => "30-553",
                                    "country" => "POL"
                                                                )
                        )

            );



        $customer_base64 = base64_encode(json_encode($customer));

?>
```

[24] https://en.wikipedia.org/wiki/JSON
[25] https://pl.wikipedia.org/wiki/Base64

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 nieznany 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*

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/

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/ - adres formularza płatności

https://ssl.dotpay.pl/test_seller/ - adres logowania do panelu administracyjnego

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 2020.

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
MasterCard	5500 0000 0000 0004	004	Tak

* przeznaczone do testów usług one-click oraz recurring

6 INFORMACJE DODATKOWE

6.1 Weryfikacja adresu IP

Adres IP serwisu Dotpay:

195.150.9.37

Z powyżej wymienionego adresu system Dotpay przesyła powiadomienia do serwisów sprzedawców. Aby komunikacja była możliwa, system sprzedawcy powinien akceptować połączenia z niniejszego adresu.

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 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 wysyła notyfikacje jest wymieniony w rozdziale *Weryfikacja adresu IP*.

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`) zależna jest 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	-	masterpass	24/7	Dostępny wyłącznie dla kont firmowych.
246	Karty płatnicze	-	VISA mastercard	24/7	Dostępny wyłącznie dla kont firmowych.

ciąg dalszy na następnej stronie

Table 17 – kontynuacja poprzedniej strony

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

ciąg dalszy na następnej stronie

Table 18 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
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	
50	Pay Way Toyota Bank	Toyota Bank Polska		24/7	
51	Płać z BOŚ	BOŚ Bank S.A.		24/7	
60	Płacę z T-Mobile Usługi Bankowe	Alior Bank S.A.Oddział T-Mobile Usługi Bankowe		24/7	
65	Płacę z Idea Bank	Idea Bank S.A.		3:00 - 22:00 / 7	
66	Bank Nowy BFG S.A.	Bankowy Fundusz Gwarancyjny (dawniej Podkarpacki Bank Spółdzielczy)		24/7	
70	Pocztowy24	Bank Pocztowy S.A.		24/7	
73	BLIK	Polski Standard Płatności Sp. z o.o.		24/7	
74	Banki Spółdzielcze	Krajowa Izba Rozliczeniowa S.A.		24/7	
75	Płacę z Plus Bank	Krajowa Izba Rozliczeniowa S.A.		24/7	

ciąg dalszy na następnej stronie

Table 18 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
76	Getin Bank PBL	Krajowa Izba Rozliczeniowa S.A.		24/7	
80	Noble Pay	Krajowa Izba Rozliczeniowa S.A.		24/7	
81	Idea Cloud	Krajowa Izba Rozliczeniowa S.A.		24/7	
83	EnveloBank	Bank Pocztowy S.A.		24/7	
84	Volkswagen Bank direct	Volkswagen Bank Polska		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 – płacę z Pl@net	Krajowa Izba Rozliczeniowa S.A.		24/7	Dostępny wyłącznie dla kont firmowych.
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.

ciąg dalszy na następnej stronie

Table 18 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
93	Kasa Stefczyka	Spółdzielcza Kasa Oszczędnościowo-Kredytowa im. F. Stefczyka	KASA STEFCZYKA	24/7	

7.1.3 PRZELEWY ONLINE

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
7	ING Klienci korporacyjni	ING Bank Śląski S.A.	ING klienci korporacyjni	pon. - sob. 8:00 - 20:00	
10	Millennium Klienci korporacyjni	Millennium Bank S.A.	Millennium bank Klienci korporacyjni	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.	CRÉDIT AGRICOLE	4:00 - 23:00 / 7	
32	BNP Paribas	BNP Paribas Bank Polska SA	BNP PARIBAS	pon. - pt. 8:00 - 21:00	
89	Santander	Santander Bank Polska SA (dawniej Bank Zachodni WBK SA)	Santander	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	

ciąg dalszy na następnej stronie

Table 20 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
21	VIA - Moje Rachunki	BillBird S.A.		pon. - pt. ok 8:00 księgowane transakcje z poprzedniego dnia	
35	Kantor Polski	Kantor Polski S.A.		24/7	
82	Przelew SEPA	-		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.		24/7	
59	CinkciarzPAY	Conotoxia Sp. z o.o.		24/7	
218	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 Gataway 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	Wymagana oddzielna umowa.

ciąg dalszy na następnej stronie

Table 24 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
95	PayPo	PayPo Sp. z o.o.	Zapłać za 30 dni PayPo	24/7	Dostępny wyłącznie dla kont firmowych dla płatności w przedziale od 40.00 PLN do 1000.00 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.	orange	24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.
232	T-Mobile	T-Mobile Polska S.A.	T	24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.
233	PLAY	P4 Sp. z o.o.	PLAY	24/7	Dostępny wyłącznie dla kont firmowych. Wymagana oddzielna umowa z partnerem Dotpay.

ciąg dalszy na następnej stronie

Table 25 – kontynuacja poprzedniej strony

NUMER	NAZWA	DOSTAWCA KANAŁU	LOGOTYP	KSIĘGOWANIE	UWAGI
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: 1.62.2.1: format zip
- Lista dostępnych kanałów płatności (polska wersja): 1.62.2.1: format xlsx
- List of available payment channels (english version): 1.62.2.1: format xlsx

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 księgowań.
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 księgowań w tym stanie. Operacja typu complaint w tym stanie oznacza rozpoczęcie procedury reklamacji.

ciąg dalszy na następnej stronie

Table 26 – kontynuacja poprzedniej strony

STATUS	ZNACZENIE / OPIS
completed (wykonana)	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. Dla typu release_rollback oznacza, że zablokowane środki z tytułu rollback wróciły na konto (`id`) sklepu. 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. Do operacji mogą pojawić się dodatkowe księgowania, ale tylko takie, które są skutkiem np. korekty, źle naliczonej korekty.
rejected (odrzucona)	Dla typów payment i payment_multimerchant_child oznacza, że Dotpay **NIE** 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. 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. Do operacji mogą pojawić się dodatkowe księgowania, ale tylko takie, które są skutkiem np. korekty, źle naliczonej korekty.
processing_realization_waiting (oczekuje na realizację)	Operacja w tym stanie oczekuje na realizację, np. dla operacji typu payout oznacza zlecenie wypłaty (ręcznie przez akceptanta, bądź przez mechanizm autowypłaty). Operacja w tym stanie oczekuje na realizację przez n dni roboczych, gdzie n wynika z podpisanej przez akceptanta umowy.
processing_realization (realizowana)	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. Gdy istnieje taka możliwość (np. kanały płatności kartami), zamiast wykonania przelewu, zwrot wykonywany jest na danym kanale.

9 DZIENNIK ZMIAN

WERSJA	DATA	OPIS ZMIAN
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 27 – 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 online) dodanie kanału 262 (Apple Pay) zmiana logotypu dla kanałów: 90 (BNP Paribas – płacę z Pl@net) rozszerzenie parametru `customer` o dodatkowe pola w Rozdziale *Obsługa danych dostawy oraz płacącego* (*account_update*, *account_update_indicator*, *fraud_activity*, *password_change*, *password_change_indicator*, *shipping_address_since_indicator*) dodanie parametu `ap_token`
1.60.16.1	2019-09-19	usunięcie kanału 33 (Volkswagen Bank)
1.59.10.2	2019-08-12	dodanie nowego parametru `order.id` w Rozdziale *Obsługa danych dostawy oraz płacącego* , dodanie opisu parametru `pid` 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: *Bezpieczeństwo integracji płatności* poprawki typograficzne
1.58.0.1	2019-05-23	poprawki typograficzne
1.56.14.1	2019-04-24	dodanie nowej wartości dla parametru `currency` : BGN, CHF, HRK, HUF, RUB
1.56.11.3	2019-04-10	usunięcie kanału 72 (Płacę z Orange) dodanie parametu `gp_token`
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 parametu `customer` oraz związanego z nim Rozdziału *Obsługa danych dostawy oraz płacącego*
1.56.11.1	2019-03-28	dodanie kanału 95 (PayPo) dodanie kanału 260 (Google Pay) dodanie aktualnej listy kanałów do pobrania w formacie xlsx 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 `operation_type` = payout_commission dodanie nowej wartości dla parametru `lang` = uk (język ukraiński) dodanie nowej wartości dla parametru `currency` = NOK wykluczenie z dokumentacji alternatywnych nazw parametrów przesyłanych do serwisu Dotpay: `kwota`, `waluta`, `opis`, `kanal`, `blokuj`, `grupykanalow`, `typ`, `txtguzik`, `data_waznosci`, `forename`, `imie`, `nazwisko`, `surname`, `ulica`, `budynek`, `lokal`, `mieszkanie`, `addr2`, `miasto`, `kod`, `telefon`, `kraj`, `jezyk`

ciąg dalszy na następnej stronie

Table 27 – kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.53.5.1	2018-12-10	poprawki typograficzne dodanie parametru `credit_card_unique_identifier` w notyfikacjach *URLC*
1.52.6.1	2018-11-13	usunięcie kanału 58 (Szybkie Płatności Internetowe z Deutsche Bank PBC) dodanie wyrażenia regularnego dla parametru `amount`
1.51.0.1	2018-10-03	dodanie parametrów `credit_card_expiration_year` i `credit_card_expiration_month` w notyfikacjach *URLC*
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 `chk` 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 `channel_groups` (`grupykanalow`) 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 `error_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 `currency` (`waluta`) dodanie nowych wartości dla parametru `language` (`jezyk`) 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 `is_completed` w notyfikacjach *URLC*
1.37.3.2	2017-06-28	dodanie podrozdziału Masscollect

ciąg dalszy na następnej stronie

Table 27 – 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 parametu 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 (grupykanalow)
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 27 – kontynuacja poprzedniej strony

WERSJA	DATA	OPIS ZMIAN
1.29.8.1	2016-02-26	dodanie parametru `channel_groups` dodatnie informacji o *HTTPS verify* oraz *SSL certificate verify* w rozdziale II. ODBIERANIE INFORMACJI PO PŁATNOŚCI (POWIADOMIENIA *URLC*) dodanie kanału 82 (Przelew SEPA) dodanie kanału 248 (Karty płatnicze) usunięcie kanału 64 (PeoPay)
1.28.5.2	2016-01-07	dodanie parametru `blik_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 *https://ssl.dotpay.pl* na *https://ssl.dotpay.pl/t2/*) 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 UNKNOWN_ACCOUNT dla parametru `error_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 `operation_withdrawal_amount`, `operation_commission_amount`, `channel_country`, `geoip_country` w notyfikacjach *URLC*
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 Pocztowy)
1.18.5.2	2014-10-28	dodanie kanałów 66 (Płacę z PBS) oraz 70 (Pocztowy24) zmiana nazwy kanału 17 (z Płać z Nordea na Płacę z IPKOnet) dodanie adresu *https://ssl.dotpay.pl/test_seller/* do rozdziału ŚRODOWISKO TESTOWE

HTTP Routing Table

/cards

```
DELETE /cards/(string:credit_card_id)/,
       47
```