
dotpay[®]

Karty

Wydanie 1.2

Dotpay Development Team

23 września, 2020

Contents

1 Adres usługi	2
2 Zasoby	2
2.1 Parametry podstawowe na wejściu metody register_order	2
2.1.1 Tabela 1. Podstawowe parametry na wejściu metody register_order	3
2.2 Parametry dla obsługi 3-D Secure v2 na wejściu metody register_order	4
2.2.1 Tabela 2. Parametry na wejściu metody register_order dla obsługi 3DS v2 opisujące przeglądarkę płacącego	5
2.2.2 Tabela 3. Obsługa danych dostawy oraz płacącego na wejściu metody register_order dla obsługi 3DS v2	6
2.2.3 Wartości używane dla pola zastępczego typu indykator dla wybranych pól:	8
2.2.4 Przykładowe żądania dla 3DS v2	9
3 Płatność One Click	11
3.1 Założenia One Click	11
3.2 Schemat procesu pierwszej płatności One Click	12
3.2.1 Rejestracja bezpośrednia	12
3.2.2 Rejestracja przy płatności	13
3.3 Opis procesu pierwszej płatności One Click	13
3.4 Schemat procesu kolejnej płatności One Click	15
3.5 Opis procesu kolejnej płatności One Click	15
4 Płatności cykliczne (recurring)	16
4.1 Płatności cykliczne - Założenia	16
4.2 Schemat procesu pierwszej płatności cyklicznej	16
4.3 Schemat procesu kolejnej płatności cyklicznej	17
4.4 Opis procesu kolejnej płatności cyklicznej	17
5 Obsługa 3-D Secure (redirect)	18
6 Informacje dodatkowe	19
6.1 Wyrejestrowanie zapisanej karty	19
7 Środowisko testowe	19

Dokument opisuje wytyczne dla sklepów internetowych (sprzedawców) wdrażających obsługę płatności kartami poprzez bezpośrednią komunikację z API RESTowym Dotpay.

Ta dokumentacja dostępna jest online pod adresem: <https://www.dotpay.pl/developer/doc/credit-cards/>

1 Adres usługi

Usługa dostępna jest pod następującymi adresami:

- dla środowiska testowego
`https://ssl.dotpay.pl/test_payment/payment_api/v1/`
- dla środowiska produkcyjnego
`https://ssl.dotpay.pl/t2/payment_api/v1/`

2 Zasoby

POST /register_order/

Metoda pozwala na rejestrację płatności w systemie Dotpay na dowolnym kanale płatności. Poniższe przykłady dotyczą wykorzystania metody do rejestracji płatności na kanale kart płatniczych.

Przykładowy request:

```
curl --user login:passwd \  
-H'Accept: application/json; indent=4' \  
-H'content-type: application/json' \  
-XPOST \  
-d @request.json \  
https://ssl.dotpay.pl/test_payment/payment_api/v1/register_order/
```

Status Codes

- 201 Created – utworzono
- 400 Bad Request – błąd podczas obsługi requestu

2.1 Parametry podstawowe na wejściu metody register_order

Strukturę danych przekazywanych na wejściu metody register_order opisuje poniższa tabela.

2.1.1 Tabela 1. Podstawowe parametry na wejściu metody register_order

Element	Typ	Uwagi
order	object	wymagane; dane zamówienia
order.amount	decimal(10,2)	wymagane; kwota zamówienia
order.currency	string	wymagane; trzyliterowy kod (ISO 4217) waluty zamówienia
order.description	string	wymagane; opis zamówienia
order.control	string	opcjonalne; identyfikator zamówienia po stronie sklepu
seller	object	wymagane; dane sklepu, na którym dokonywana jest płatność
seller.account_id	integer	wymagane; numer konta Dotpay
seller.url	string	wymagane; adres na jaki płacący może zostać przekierowany po zakończeniu płatności
seller.urlc	string	opcjonalne; adres na jaki będą przesyłane notyfikacje o zmianach statusu operacji
payer	object	wymagane; dane płatcego
payer.first_name	string	wymagane; imię płatcego
payer.last_name	string	wymagane; nazwisko płatcego
payer.email	string	wymagane; adres email płatcego
payer.address	object	opcjonalne (chyba, że konfiguracja danego kanału wymaga podania tych danych); dane adresowe płatcego
payer.address.street	string	wymagane w przypadku podania payer.address; ulica
payer.address.building_number	string	wymagane w przypadku podania payer.address; numer budynku
payer.address.flat_number	string	opcjonalne w przypadku podania payer.address; numer mieszkania
payer.address.postcode	string	wymagane w przypadku podania payer.address; kod pocztowy
payer.address.city	string	wymagane w przypadku podania payer.address; miasto
payer.address.country	string	wymagane w przypadku podania payer.address; trzyliterowy kod kraju (ISO 3166-1 alpha-3)
payment_method	object	wymagane; dane metody płatności
payment_method.channel_id	integer	wymagane; numer kanału płatności, dla kart płatniczych jest to 248. Pełna lista jest dostępna w podstawowej Dokumentacji technicznej implementacji płatności
payment_method.credit_card	object	dane karty płatniczej
payment_method.credit_card.number	string	numer karty
payment_method.credit_card.expiration_date	object	data ważności karty
payment_method.credit_card.expiration_date.year	string (YYYY)	rok ważności karty
payment_method.credit_card.expiration_date.month	string (MM)	miesiąc ważności karty
payment_method.credit_card.security_code	string	kod CVV2/CVC2
payment_method.credit_card.store	boolean	zgoda na zarejestrowanie danych karty po stronie Dotpay

ciąg dalszy na następnej stronie

Table 1 – kontynuacja poprzedniej strony

Element	Typ	Uwagi
payment_method.credit_card.customer_id	string (4 - 1024 znaki)	unikalny identyfikator płatącego nadany i przechowywany przez system sprzedawcy, wymagany podczas kolejnych płatności
payment_method.credit_card.id	string	identyfikator karty klienta zarejestrowanej po stronie systemu Dotpay
payment_method.credit_card.operation_type	string	typ operacji: e_commerce – pierwsza oraz kolejna płatność w modelu one-click (wartość domyślna), recurring_init – pierwsza transakcja, umożliwiającą późniejsze skorzystanie z płatności cyklicznych, recurring – płatność cykliczna (klient nie musi być obecny podczas obciążenia zapisanej wcześniej karty),
payment_method.credit_card.security_code_required	string	wymagalność kodu zabezpieczającego CVV/CVV2, dotyczy jedynie kolejnej operacji e_commerce. Dostępne wartości: yes (domyślnie) no
payment_method.credit_card.threeds	string	wymagalność kodu autoryzacyjnego 3-D Secure. Dotyczy jedynie operacji e_commerce dla kart uczestniczących w programie. Dostępne wartości: yes (domyślnie) no
request_context.ip	string	wymagane; adres ip płatącego
request_context.language	string	dwuliterowy kod języka (ISO 639-1) w jakim wykonywana jest płatność; pl (domyślnie)

2.2 Parametry dla obsługi 3-D Secure v2 na wejściu metody register_order

Przesłanie większej liczby danych niż tylko „wymagane” przy płatności kartowej, może mieć duże znaczenie przy ostatecznej decyzji wydawcy karty dotyczącej akceptacji samej transakcji.

Informacja: Na podstawie przesłanych dodatkowych informacji lub ich braku wydawca karty może zdecydować o ewentualnej konieczności dodatkowej weryfikacji transakcji (challenge) lub procesowaniu transakcji bez kodu 3DS. To z kolei może przyspieszyć i ułatwić sam proces płatności dla płatącego a w konsekwencji pozytywnie wpłynąć na konwersję płatności kartowych.

Dlatego, o ile to możliwe zalecamy wysłanie jak największej ilości dodatkowych danych podczas inicjalizacji płatności.

Strukturę danych przekazywanych na wejściu metody register_order dla obsługi **3DS v2** opisują poniższe tabele.

2.2.1 Tabela 2. Parametry na wejściu metody register_order dla obsługi 3DS v2 opisujące przeglądarkę płatącego

Element	Typ	Uwagi
<code>request_context.accept</code>	string	zalecane; Nagłówek Accept z nagłówków przeglądarki klienta opis: HTTP ACCEPT Przykład: <code>request_context.accept = text/html, application/xhtml+xml, application/xml;q=0.9, */</code>
<code>request_context.referer</code>	string	zalecane; Adres strony z której użytkownik został przekierowany (nagłówek HTTP) opis: HTTP referer Przykład: <code>request_context.referer = http://www.example.org/referring_page</code>
<code>request_context.useragent</code>	string	zalecane; Nagłówek user-agent z nagłówków przeglądarki klienta opis: HTTP User-Agent Przykład: <code>request_context.useragent = Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36</code>
<code>request_context.browser.javaenabled</code>	boolean	zalecane; Możliwość wykonywania kodu Java w przeglądarce klienta opis: navigator.javaEnabled() ; Przykład: <code>request_context.browser.javaenabled = 1</code>
<code>request_context.browser.javascriptenabled</code>	boolean	zalecane; Możliwość wykonywania kodu JavaScript w przeglądarce klienta Przykład: <code>request_context.browser.javascriptenabled = 1</code>
<code>request_context.browser.language</code>	string	wymagane gdy <code>request_context.browser.javascriptenabled = 1</code> Język przeglądarki w standardzie IETF BCP 47 opis: navigator.language.slice(0,2) Przykład: <code>request_context.browser.language = pl</code>
<code>request_context.browser.screencolordepth</code>	int	wymagane gdy <code>request_context.browser.javascriptenabled = 1</code> Głębokość koloru dla wyświetlania koloru w przeglądarce klienta, pozyskana z <code>screen.colorDepth</code> . opis: screen.colorDepth dopuszczalne wartości: <code>1,4,8,15,16,24,32,48</code> Przykład: <code>request_context.browser.screencolordepth = 24</code>

ciąg dalszy na następnej stronie

Table 2 – kontynuacja poprzedniej strony

Element	Typ	Uwagi
<code>request_context.browser.screenheight</code>	int	wymagane gdy <code>request_context.browser.javascriptenabled = 1</code> Wysokość ekranu w pikselach pozyskana z <code>screen.height</code> . opis: <code>screen.height</code> Przykład: <code>request_context.browser.screenheight = 1080</code>
<code>request_context.browser.screenwidth</code>	int	wymagane gdy <code>request_context.browser.javascriptenabled = 1</code> Szerokość ekranu w pikselach pozyskana z <code>screen.width</code> . opis: <code>screen.width</code> Przykład: <code>request_context.browser.screenwidth = 1920</code>
<code>request_context.browser.timezone</code>	int	wymagane gdy <code>request_context.browser.javascriptenabled = 1</code> Strefa czasowa wyrażona jako różnica w minutach pomiędzy czasem GMT a czasem lokalnym opis: <code>new Date().getTimezoneOffset()</code> Przykład: <code>request_context.browser.timezone = -120</code>

2.2.2 Tabela 3. Obsługa danych dostawy oraz płacącego na wejściu metody `register_order` dla obsługi 3DS v2

NAZWA POLA	TYP	OPIS
<code>payment_method.customer.is_logged_in</code>	boolean	Informacja czy płacący przed dokonaniem płatności był zarejestrowany w systemie sprzedawcy
<code>payment_method.customer.registered_since</code>	string	Data rejestracji płacącego w serwisie sprzedawcy, format <code>YYYY-MM-DD</code> lub <code>YYYY-MM-DD hh:mm:ss</code> Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: <code>payment_method.customer.order_count</code> . Zamiast podawania konkretnej daty w formacie <code>YYYY-MM-DD</code> można użyć zastępczo parametru: <code>payment_method.customer.registered_since_indicator</code> .
<code>payment_method.customer.registered_since_indicator</code>	string (indykator)	Data rejestracji płacącego w serwisie sprzedawcy, indyikator dla pola <code>payment_method.customer.registered_since</code> Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr: <code>payment_method.customer.order_count</code>
<code>payment_method.customer.account_update</code>	string	Data ostatniej zmiany konta płacącego w serwisie sprzedawcy, format <code>YYYY-MM-DD</code> Zamiast podawania konkretnej daty w formacie <code>YYYY-MM-DD</code> można użyć zastępczo parametru: <code>payment_method.customer.account_update_indicator</code> .
<code>payment_method.customer.account_update_indicator</code>	string (indykator)	Data ostatniej zmiany konta płacącego w serwisie sprzedawcy, indyikator dla pola <code>payment_method.customer.account_update</code>

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

NAZWA POLA	TYP	OPIS
payment_method. customer. password_change	string	Data ostatniej zmiany hasła dla konta płacącego w serwisie sprzedawcy, format YYYY-MM-DD Zamiast podawania konkretnej daty w formacie YYYY-MM-DD można użyć zastępczo parametru: payment_method.customer.password_change_indicator.
payment_method. customer. password_change_indicator	string (in- dykator)	Data ostatniej zmiany hasła dla konta płacącego w serwisie sprzedawcy, indyktor dla pola payment_method.customer.password_change
payment_method. customer. shipping_address_since	string	Data od kiedy podany adres dostawy płacącego w serwisie sprzedawcy jest używany, format YYYY-MM-DD Zamiast podawania konkretnej daty w formacie YYYY-MM-DD można użyć zastępczo parametru: payment_method.customer.shipping_address_since_indicator.
payment_method. customer. shipping_address_since_indicator	string (in- dykator)	Data od kiedy podany adres dostawy płacącego w serwisie sprzedawcy jest używany, indyktor dla pola payment_method.customer.shipping_address_since
payment_method. customer.order_count	int	Ilość złożonych zamówień przez płacącego w serwisie sprzedawcy od daty rejestracji Pole opcjonalne, jeżeli jest wysyłane to należy również wysłać parametr payment_method.customer.registered_since
payment_method. customer. order_count_day	int	Ilość złożonych zamówień przez płacącego w serwisie sprzedawcy w tym samym dniu
payment_method. customer. order_count_year	int	Ilość złożonych zamówień przez płacącego w serwisie sprzedawcy w tym samym roku
payment_method. customer.fraud_activity	boolean	Czy sklep kiedykolwiek zanotował podejrzaną aktywność na koncie tego kupującego
payment_method. customer.order	-	Zamówienie
payment_method. customer.order. total_amount	string	Wartość całego zamówienia
payment_method. customer.order.id	string	Identyfikator zamówienia w systemie sprzedawcy. Odpowiada numerowi ID całego zamówienia w bazie sklepu
payment_method. customer.order. delivery_type	string	Metoda dostawy Dostępne wartości: <ul style="list-style-type: none"> • 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)
payment_method. customer.order. delivery_address	-	Adres dostawy Jeśli paczka jest dostarczana do punktu/paczkomatu/itd, to taki adres i nazwa powinna być podana a nie dane faktycznego odbiorcy.
payment_method. customer.order. delivery_address.city	string	Adres dostawy: miasto
payment_method. customer.order. delivery_address.street	string	Adres dostawy: ulica

ciąg dalszy na następnej stronie

Table 3 – kontynuacja poprzedniej strony

NAZWA POLA	TYP	OPIS
payment_method. customer.order. delivery_address. building_number	string	Adres dostawy: numer budynku
payment_method. customer.order. delivery_address. flat_number	string	Adres dostawy: numer mieszkania
payment_method. customer.order. delivery_address. postcode	string	Adres dostawy: kod pocztowy
payment_method. customer.order. delivery_address. country	string	Adres dostawy: dwuliterowy (ISO 3166-1 alpha2) lub trzyliterowy (ISO 3166-1 alpha3) kod kraju
payment_method. customer.order. delivery_address.name	string	Nazwa odbiorcy/punktu odbiorczego. Przykłady: payment_method.customer.order. delivery_address.name = Point PP:6252652 payment_method.customer.order. delivery_address.name = PPP:6252652
payment_method. customer.order. delivery_address.phone	string	Numer telefonu odbiorcy
payment_method. customer.order. delivery_address. is_verified	bool	Adres dostawy: czy adres dostawy jest zweryfikowany

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

2.2.3 Wartości używane dla pola zastępczego typu indykator dla wybranych pól:

WARTOŚĆ	OPIS
01	Konto płaćącego 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

2.2.4 Przykładowe żądania dla 3DS v2

Poniżej przykładowe żądania z użyciem powyższych parametrów:

Listing 1: Przykład 1: użycie minimalnej ilości parametrów dla procesu 3DS v2 (format json)

```
1 {
2   "order": {
3     "amount": "34.00",
4     "currency": "PLN",
5     "description": "Payment for order no 3342",
6     "control": "xcftg-32432-5325hdf"
7   },
8   "seller": {
9     "account_id": "123456",
10    "url": "https://www.example.com"
11  },
12  "payer": {
13    "first_name": "John",
14    "last_name": "Doe",
15    "email": "johndoemail@example.com",
16    "phone": "123456789",
17    "address": {
18      "city": "Warszawa",
19      "street": "Krucza",
20      "building_number": "1a",
21      "flat_number": "4",
22      "postcode": "00-950",
23      "country": "PL"
24    }
25  },
26  "payment_method": {
27    "channel_id": "248",
28    "credit_card": {
29      "number": "4929532027887670",
30      "expiration_date": {
31        "year": "2022",
32        "month": "01"
33      },
34      "security_code": "670",
35      "store": "1",
36      "customer_id": "f9c6a4-25473-765gh"
37    }
38  },
39  "request_context": {
40    "ip": "127.0.0.1",
41    "language": "pl",
42    "accept": "text/html, application/xhtml+xml, application/xml;q=0.9, */",
43    "referer": "http://www.example.org/referring_page",
44    "useragent": "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, ↵
↳like Gecko) Chrome/84.0.4147.105 Safari/537.36",
45    "browser": {
46      "javaenabled": 1,
47      "javascriptenabled": 1,
48      "language": "en",
49      "screencolordepth": 24,
50      "screenheight": 1024,
51      "screenwidth": 1920,
52      "timezone": -120
53    }
54  }
55 }
```

(ciąg dalszy na następnej stronie)

55 }
56 }

Listing 2: Przykład 2: użycie dodatkowych parametrów dla procesu 3DS v2 - płatność typu one-click z zapisaną wcześniej kartą (format json)

```

1 {
2   "order": {
3     "amount": "56.20",
4     "currency": "PLN",
5     "description": "Payment for order no 6542",
6     "control": "3426hs5fskdbg6g"
7   },
8   "seller": {
9     "account_id": "123456",
10    "url": "https://www.example.com"
11  },
12  "payment_method": {
13    "channel_id": "248",
14    "credit_card": {
15      "id":
↪"85c14e6e5608cbc69e19acec41730d59052fbc306364d96c9cdaafacb7a0833d0fa14280ab9a2b2381fad381f65f0
↪",
16      "customer_id": "f9c6a4-25473-765gh"
17    },
18  },
19  "customer": {
20
21    "is_logged_in": 1,
22    "registered_since": "2019-11-21",
23    "order_count": 23,
24
25    "order": {
26      "id": "54356723",
27      "delivery_type": "PICKUP_POINT",
28      "delivery_address": {
29        "name": "Point PP:6252652",
30        "phone": "+48987654321",
31        "street": "Zielona",
32        "building_number": "32",
33        "postcode": "61-321",
34        "city": "Konin",
35        "country": "PL",
36        "is_verified": 1
37      }
38    },
39    "payer": {
40      "first_name": "Wieslaw",
41      "last_name": "Nowak",
42      "email": "paysdfds@example.com",
43      "phone": "+48443456766"
44    }
45  },
46
47  },
48  "payer": {
49    "first_name": "Adam",
50    "last_name": "Kowal",
51    "email": "payeremail@example.com",

```

(ciąg dalszy na następnej stronie)

```

52     "phone": "+48123456789",
53     "address": {
54         "city": "Konin",
55         "street": "Prosta",
56         "building_number": "34",
57         "flat_number": "7",
58         "postcode": "62-500",
59         "country": "PL"
60     }
61 },
62 "request_context": {
63     "ip": "192.188.3.221",
64     "language": "pl",
65     "accept": "text/html, application/xhtml+xml, application/xml;q=0.9, */",
66     "referer": "http://www.example.org/referring_page",
67     "useragent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36_
→(KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36 Edge/14.14393",
68     "browser": {
69         "javaenabled": 1,
70         "javascriptenabled": 1,
71         "language": "en",
72         "screencolordepth": 24,
73         "screenheight": 1024,
74         "screenwidth": 1920,
75         "timezone": -120
76     }
77 }
78 }
79 }

```

3 Płatność One Click

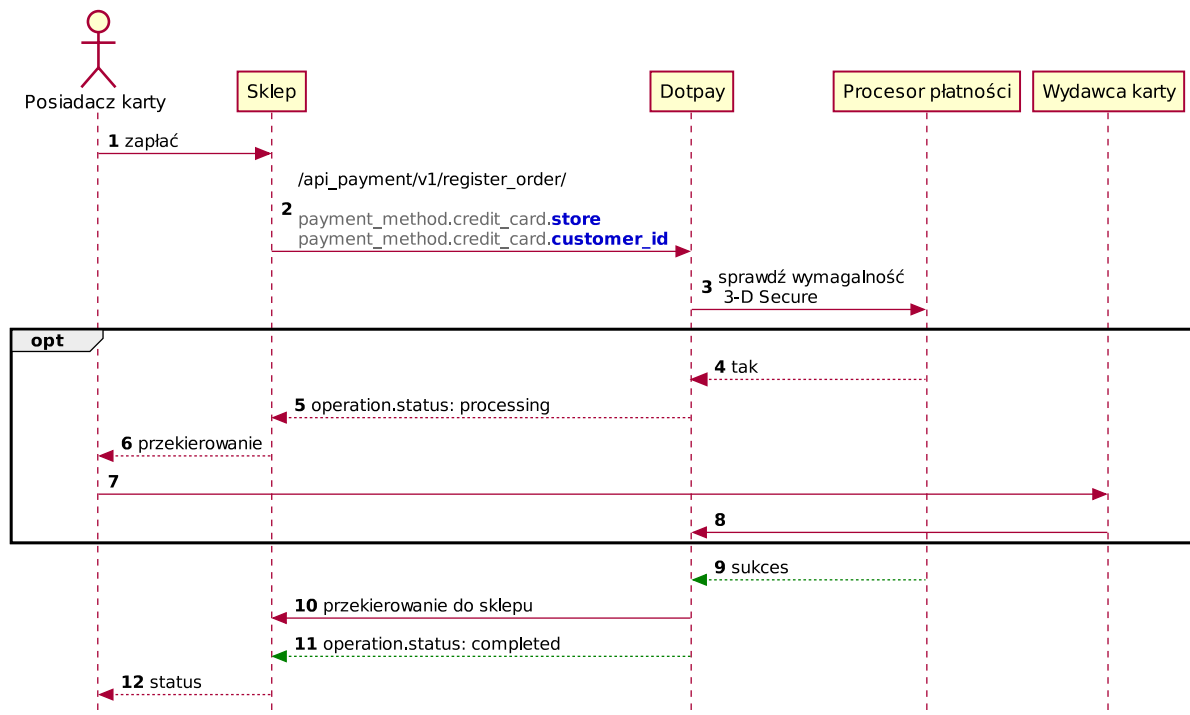
3.1 Założenia One Click

Niniejsza sekcja opisuje przykładowy proces rejestracji (bezpośredniej i pośredniej) karty w modelu One Click, oraz proces kolejnych płatności, gdzie sklep przekazuje identyfikator wcześniej zarejestrowanej w Dotpay karty.

Sklep może wysyłać żądania zawierające token karty wyłącznie wtedy, gdy pochodzą one od klientów uwierzytelnionych w systemie sklepu (klient musi być zalogowany).

Ostrzeżenie: Należy pamiętać, że karty są rejestrowane w kontekście danej grupy sklepów (id) w Dotpay i nie zadziałają dla innych kont.

3.2 Schemat procesu pierwszej płatności One Click



Poniżej zostały przedstawione przykłady inicjalizacji pierwszej płatności w danym modelu:

3.2.1 Rejestracja bezpośrednia

POST /cards/

```

{
  "seller": {
    "account_id": "123456",
    "url": "https://www.example.com"
  },
  "payer": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoemail@example.com"
  },
  "credit_card": {
    "number": "4929532027887670",
    "expiration_date": {
      "year": "2020",
      "month": "01"
    },
    "security_code": "670",
    "customer_id": "f9c6a4-25473"
  },
  "request_context": {
    "ip": "127.0.0.1",
    "language": "pl"
  }
}

```

3.2.2 Rejestracja przy płatności

POST /register_order/

```
{
  "order": {
    "amount": "1.00",
    "currency": "PLN",
    "description": "test",
    "control": "test"
  },
  "seller": {
    "account_id": "123456",
    "url": "https://www.example.com"
  },
  "payer": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoemail@example.com"
  },
  "payment_method": {
    "channel_id": "248",
    "credit_card": {
      "number": "4929532027887670",
      "expiration_date": {
        "year": "2020",
        "month": "01"
      },
      "security_code": "670",
      "store": "1",
      "customer_id": "f9c6a4-25473"
    },
    "request_context": {
      "ip": "127.0.0.1",
      "language": "pl"
    }
  }
}
```

3.3 Opis procesu pierwszej płatności One Click

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).

Alternatywnie karta może być zarejestrowana na zasadzie przekierowania do serwisu Dotpay, gdzie klient bezpiecznie poda wszystkie dane kartowe. Opis tego procesu można znaleźć w [Dokumentacji technicznej implementacji płatności](#)

Poniższy opis dotyczy rejestracji karty wraz z płatnością. Przy rejestracji bezpośredniej proces jest identyczny, przy czym zamiast obciążenia karty nastąpi chwilowa blokada środków na czas rejestracji w systemie, która zostanie automatycznie zwolniona po jej zakończeniu. Typ operacji zmieni się też z payment na credit_card_registration.

1. Kupujący wybiera płatność kartą, podaje jej dane i klika zapłacić.
2. Sklep inicjalizuje proces płatności w Dotpay przekazując dane zamówienia takie jak dane karty oraz parametry wymagane do jej rejestracji, przykładowo:

```

{
  "order": {
    "amount": "1.00",
    "currency": "PLN",
    "description": "test",
    "control": "test"
  },
  "seller": {
    "account_id": "123456",
    "url": "https://www.example.com"
  },
  "payer": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoemail@example.com"
  },
  "payment_method": {
    "channel_id": "248",
    "credit_card": {
      "number": "4929532027887670",
      "expiration_date": {
        "year": "2020",
        "month": "01"
      },
      "security_code": "670",
      "store": "1",
      "customer_id": "f9c6a4-25473"
    },
    "request_context": {
      "ip": "127.0.0.1",
      "language": "pl"
    }
  }
}

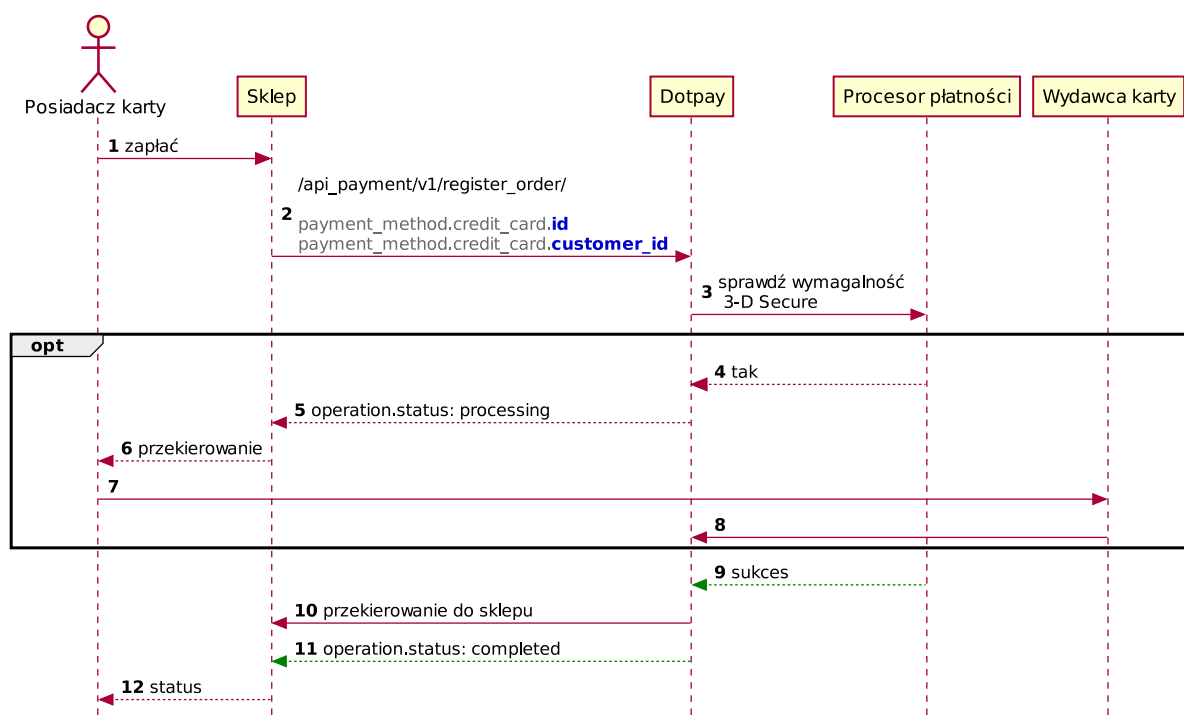
```

3. System Dotpay odpytuje o to, czy karta uczestniczy w programie 3-D Secure.

Uwaga: Krok 4-8 może być opcjonalny, jeżeli dana karta uczestniczy w programie 3-D Secure (opis schematu w *Rozdziale 6*).

4. Jeżeli uczestniczy,
5. system Dotpay zwraca szczegóły operacji wraz z sekcją `redirect` i adres `redirect_simplified_url`.
6. Sklep odpowiedzialny jest za przekierowanie płacącego na strony wydawcy bezpośrednio (obsługa sekcji `redirect`) bądź pośrednio przez Dotpay (przekierowanie na adres `redirect_simplified_url`).
7. Płacący przechodzi na strony wydawcy, na których uwierzytelnia się mechanizmem 3-D Secure.
8. Wydawca przekierowuje płacącego na strony Dotpay.
9. Następuje obciążenie karty oraz jej rejestracja
10. Płacący jest przekierowywany na strony sklepu.
11. Po odebraniu notyfikacji urlc o statusie operacji
12. sklep informuje kupującego o statusie zamówienia.

3.4 Schemat procesu kolejnej płatności One Click



3.5 Opis procesu kolejnej płatności One Click

1. Kupujący, po wybraniu kanału płatności, wybiera zarejestrowaną kartę i klika zapłać.
2. Sklep inicjalizuje proces płatności w Dotpay przekazując dane zamówienia wraz z identyfikatorem zarejestrowanej karty (oraz znanym sobie `customer_id`) przykładowo:

```

{
  "order": {
    "amount": "1.00",
    "currency": "PLN",
    "description": "test",
    "control": "test"
  },
  "seller": {
    "account_id": "123456",
    "url": "https://www.example.com"
  },
  "payer": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoemail@example.com"
  },
  "payment_method": {
    "channel_id": "248",
    "credit_card": {
      "id":
↪"85c14e6e5608cbc69e19acec41730d59052fbc306364d96c9cdaafacb7a0833d0fa14280ab9a2b2381fad381f65f0
↪",
      "customer_id": "f9c6a4-25473"
    }
  },
  "request_context": {

```

(ciąg dalszy na następnej stronie)

```
        "ip": "127.0.0.1",  
        "language": "pl"  
    }  
}
```

3. System Dotpay odpytuje o to, czy karta uczestniczy w programie 3-D Secure.

Uwaga: Krok 4-8 może być opcjonalny, jeżeli dana karta uczestniczy w programie 3-D Secure.

4. Jeżeli uczestniczy,
5. system Dotpay zwraca szczegóły operacji wraz z sekcją `redirect` i adres `redirect_simplified_url`.
6. Sklep odpowiedzialny jest za przekierowanie płatącego na strony wydawcy bezpośrednio (obsługa sekcji `redirect`) bądź pośrednio przez Dotpay (przekierowanie na adres `redirect_simplified_url`).
7. Płacący przechodzi na strony wydawcy, na których uwierzytelnia się mechanizmem 3-D Secure.
8. Wydawca przekierowuje płatącego na strony Dotpay.
9. Następuje obciążenie karty.
10. Płacący jest przekierowywany na strony sklepu.
11. Po odebraniu notyfikacji urlc o statusie operacji
12. sklep informuje kupującego o statusie zamówienia.

4 Płatności cykliczne (recurring)

4.1 Płatności cykliczne - Założenia

Niniejsza sekcja opisuje przykładowy proces rejestracji (bezpośredniej i pośredniej) karty w modelu Recurring, oraz proces kolejnych płatności, gdzie sklep inicjuje płatność bez udziału klienta, przekazując identyfikator wcześniej zarejestrowanej w Dotpay karty.

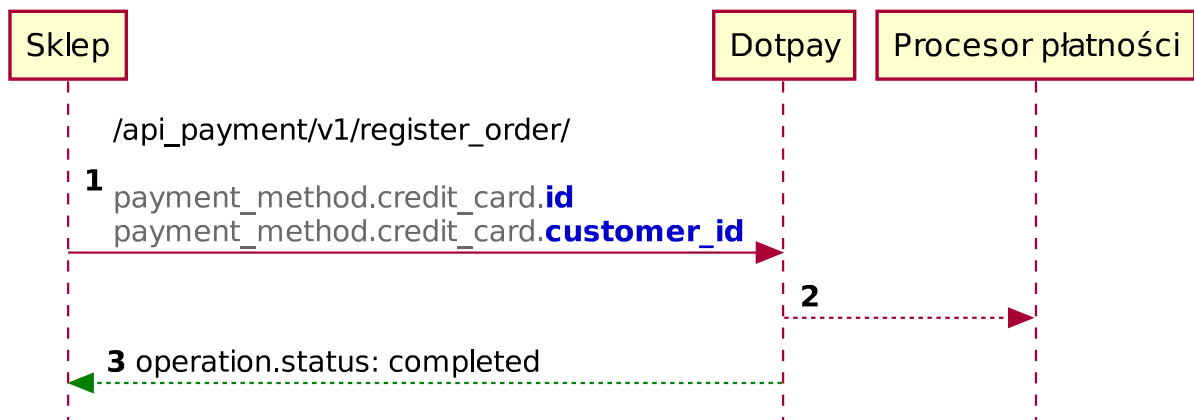
Ostrzeżenie: Należy pamiętać, że karty są rejestrowane w kontekście danej grupy sklepów (`id`) w Dotpay i nie zadziałają dla innych kont.

4.2 Schemat procesu pierwszej płatności cyklicznej

Schemat jest analogiczny jak w przypadku pierwszej płatności One Click. Jedynie (w zależności od ustawień konta) musi być dodatkowo przekazany parametr `payment_method.credit_card.operation_type = recurring_init`.

Ostrzeżenie: Poprawna rejestracja karty nie gwarantuje sukcesu przy próbie późniejszego obciążenia. Klient może w dowolnym wyrejestrować daną kartę, bądź transakcja nie powiedzie się ze względu na niewystarczającą ilość środków, limity dzienne, odmowę autoryzacji wydawcy karty itp..

4.3 Schemat procesu kolejnej płatności cyklicznej



4.4 Opis procesu kolejnej płatności cyklicznej

1. Sklep inicjalizuje proces płatności w Dotpay przekazując dane zamówienia wraz z identyfikatorem zarejestrowanej karty (oraz znanym sobie `customer_id`) przykładowo:

```
{
  "order": {
    "amount": "1.00",
    "currency": "PLN",
    "description": "test",
    "control": "test"
  },
  "seller": {
    "account_id": "123456",
    "url": "https://www.example.com"
  },
  "payer": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoemail@example.com"
  },
  "payment_method": {
    "channel_id": "248",
    "credit_card": {
      "id":
↪"85c14e6e5608cbc69e19acec41730d59052fbcd306364d96c9cdaafac7a0833d0fa14280ab9a2b2381fad381f65f0"
↪",
      "customer_id": "f9c6a4-25473"
    }
  },
  "request_context": {
    "ip": "127.0.0.1",
    "language": "pl"
  }
}
```

2. Następuje obciążenie karty
3. i Dotpay wysyła notyfikację urlc o wykonanej operacji.

Ostrzeżenie: W przypadku odmowy autoryzacji płatności dla karty uprzednio zarejestrowanej, kolejna próba obciążenia powinna być wykonana nie wcześniej niż w kole-

nym dniu i nie częściej niż raz dziennie przez okres nie dłuższy niż 31 dni. W tym czasie Sprzedawca powinien podjąć działania zmierzające do ustalenia z Klientem problemów z obciążeniem karty.

5 Obsługa 3-D Secure (redirect)

Jeśli dalsze procesowanie płatności wymaga przekierowania płatącego do banku / wydawcy, w odpowiedzi Dotpay zwróci dodatkowo obiekt `redirect` zgodnie z poniższym opisem.

Element	Typ	Uwagi
<code>redirect</code>	object	komplet danych potrzebny do przekierowania płatącego do banku / wydawcy
<code>redirect.url</code>	string	adres url na jaki należy przekierować płatącego
<code>redirect.method</code>	enumeration (post, get)	metoda http jaką należy przekierować płatącego do banku / wydawcy
<code>redirect.data</code>	object	słownik (lista par <klucz, wartość>) parametrów, z którymi należy przekierować płatącego do banku / wydawcy
<code>redirect.encoding</code>	string	strona kodowa do jakiej należy przekonwertować wartości ze słownika <code>request.data</code> z kodowania utf-8 przed wysłaniem do banku / wydawcy

Uwaga: Dane, z którymi należy przekierować płatącego do banku / wydawcy, zawierają sygnaturę zapewniającą ochronę integralności danych. Muszą zatem zostać przekazane w formie niezmienionej (z dokładnością do konwersji do odpowiedniej strony kodowej). Jeśli integralność danych zostanie naruszona, płatność zostanie odrzucona po stronie banku / wydawcy.

Informacja: Jako alternatywne podejście można zastosować przekierowanie (HTTP 302) na adres zwrócony w parametrze `redirect_simplified_url`. Przekierowanie do banku / wydawcy zostanie wtedy wykonane po stronie Dotpay.

Listing 3: Przykładowa odpowiedź z Dotpay przy próbie rejestracji karty zawierająca adresy `redirect.url` oraz `redirect_simplified_url`:

```
{  
  
  "redirect": {  
    "url": "https://ssl.dotpay.pl/test_payment/channel_specific/pv/payment_  
↪authentication/M1234-56789/  
↪k5bd2c03b5d995boe1862cf775cf8cec114fe36aea928272b0a2b4883a92b14d/",  
    "data": {},  
    "method": "GET",  
    "encoding": "utf-8"  
  }  
}
```

(ciąg dalszy na następnej stronie)

```

    },
    "redirect_simplified_url": "https://ssl.dotpay.pl/test_payment/channel_
↪specific/pv/payment_authentication/M1234-56789/
↪k5bd2c03b5d995boe1862cf775cf8cec114fe36aea928272b0a2b4883a92b14d/"
  }

```

6 Informacje dodatkowe

6.1 Wyrejestrowanie zapisanej karty

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.

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	Usunięto
404 Not Found	Nie znaleziono karty
400 Bad Request	Błąd podczas obsługi żądania

7 Środowisko testowe

Poniżej znajduje się lista przykładowych kart, które można wykorzystać podczas testów na środowisku testowym Dotpay. Data ważności musi być przyszła, ale nie późniejsza niż grudzień 2020.

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

HTTP Routing Table

/cards

POST /cards/, 12

DELETE /cards/(string:credit_card_id)/,
19

/register_order

POST /register_order/, 2