

dpd

# Info Services

## Spis treści

1	Wstęp .....	3
1.1	Cel dokumentu .....	3
1.2	Słownik pojęć.....	3
1.3	Definicja uprawnień i kanałów .....	3
1.4	Adresy usługi .....	4
1.5	Standard bezpieczeństwa .....	4
2	Opis biznesowy .....	4
2.1	Kolejność wykonywanych operacji .....	4
2.2	Ograniczenia w ilości wysyłanych danych .....	5
2.3	Ogólny opis interfejsów i ich różnic .....	5
2.3.1	Interfejsy DPDInfoServicesXmlEvents oraz DPDInfoServicesObjEvents .....	5
3	Opis struktury serwisu .....	8
3.1	Autoryzacja .....	9
3.1.1	DPDInfoServicesObjEvents i DPDInfoServicesXmlEvents .....	9
3.2	Opis metod interfejsów .....	10
3.2.1	Interfejs DPDInfoServicesObjEvents .....	10
3.2.2	Interfejs DPDInfoServicesXmlEvents .....	24
3.3	Kody błędów .....	38
4	Przekierowania i zwroty .....	38

# 1 Wstęp

## 1.1 Cel dokumentu

Celem dokumentu jest opisanie sposobu działania i wykorzystania interfejsów typu WebService do komunikacji systemów klienckich z systemami DPD.

Dostępne są następujące usługi:

- pobieranie listy zdarzeń dla paczek
- potwierdzenie odebrania listy zdarzeń
- pobranie podpisu klienta

Każda usługa jest autoryzowana loginem i hasłem.

Klient korzystający z przedstawionego rozwiązania w celu integracji własnych systemów z systemami DPD Polska ma pewność co do zgodności przygotowanych przesyłek z obowiązującym standardem, co przekłada się na poprawę wydajności i niezawodności naszych usług.

## 1.2 Słownik pojęć

Pojęcie	Definicja
Numer listu przewozowego	Numer identyfikujący daną paczkę; inaczej: waybill.
Zdarzenie	Fakt zaistniały w związku z cyklem życia paczki lub przesyłki (np. nadanie, doręczenie).
Paczka	Paczka jest elementem składowym przesyłki. Paczka posiada określone wymiary (długość, szerokość, wysokość, wagę). Paczka identyfikowana jest za pomocą numeru listu przewozowego.

## 1.3 Definicja uprawnień i kanałów

Pełna konfiguracja konta klienta upoważnionego do korzystania z usług webserwisowych odbywa się po stronie systemów DPD Polska po wcześniejszych ustaleniach z klientem. Klient pobiera zdarzenia z przypisanego mu kanału informacyjnego. Kanał informacyjny definiowany jest poprzez numer płatnika (jako parametr wywołania API) oraz listę podrzędnych numerów klienta, na których przesyłki zostały wygenerowane.

Zdarzenie jest dostępne w danym kanale informacyjnym jeśli została ona utworzona na którymś z przydzielonych do kanału numkatów (numerów podrzędnych).

Jedno konto może mieć wiele kanałów informacyjnych. Jeden kanał informacyjny może być przypisany maksymalnie do jednego konta.

Klient ma również do dyspozycji metodę, która nie wykorzystuje kanału (getEventsForWaybill). Po podaniu numeru listu, klient otrzymuje listę zdarzeń (lub ostatnie zdarzenie) dla paczki o podanym numerze.

## 1.4 Adresy usługi

Interfejsy dostępne są pod adresem:

Interfejs	PROD
DPDInfoServicesObjEvents	<a href="https://dpdinfoservices.dpd.com.pl/DPDInfoServicesObjEventsService/DPDInfoServicesObjEvents?wsdl">https://dpdinfoservices.dpd.com.pl/DPDInfoServicesObjEventsService/DPDInfoServicesObjEvents?wsdl</a>
DPDInfoServicesXmlEvents	<a href="https://dpdinfoservices.dpd.com.pl/DPDInfoServicesXmlEventsService/DPDInfoServicesXmlEvents?wsdl">https://dpdinfoservices.dpd.com.pl/DPDInfoServicesXmlEventsService/DPDInfoServicesXmlEvents?wsdl</a>

## 1.5 Standard bezpieczeństwa

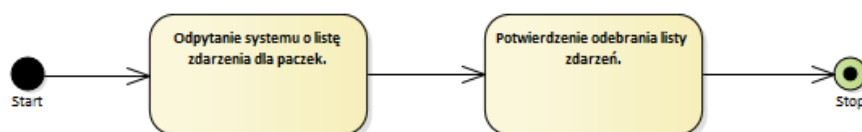
Do zabezpieczenia komunikacji używany jest standard SSL o 128-bitowym niesymetrycznym kluczu. Każde wywołanie zabezpieczone jest loginem i hasłem, jak również sprawdzane jest posiadanie przez klienta odpowiednich uprawnień (zapisanych w LDAP).

# 2 Opis biznesowy

INFOServices stanowi uniwersalne (niezależne od typu systemu) narzędzie przesyłu informacji pomiędzy systemami klienckimi a systemem DPD. Narzędzie to oparte jest o interfejsy typu WebService udostępniane dla klientów DPD.

## 2.1 Kolejność wykonywanych operacji

Poniższy diagram przedstawia kolejność wywołania operacji w ramach procesu biznesowego.



Poprawne użycie webserwisu wymaga od klienta wykonania kolejno dwóch następujących kroków:

1. odpytanie systemu o listę zdarzeń dla paczek z kanału klienckiego ( w przypadku metody `getEventsForWaybill` – bez uwzględniania kanału)
2. potwierdzenie odebrania listy zdarzeń

Brak wykonania kroku drugiego spowoduje, że przy kolejnej próbie odpytania systemu o zdarzenia (krok pierwszy) klient otrzyma w odpowiedzi ponownie te same dane.

Za pomocą webserwisu InfoServices istnieje także możliwość pobrania podpisu klienta.

## 2.2 Ograniczenia w ilości wysyłanych danych

Ilość danych zawarta w odpowiedzi do klienta jest zależna od ustawień konfiguracyjnych. Domyślnie, minimalny zbiór zwracanych zdarzeń ustawiony jest na 100 (niezależnie od parametru „limit” ustawianego w momencie wywołania metody), natomiast maksymalna ich ilość to 1000.

## 2.3 Ogólny opis interfejsów i ich różnic

INFOServices posiada cztery dostępne interfejsy, z czego dwa z nich różnią się głównie sposobem przekazywania parametrów wejściowych oraz formatem zwracanych wyników:

1. Interfejs obiektowy `DPDInfoServicesObjEvents`, który przyjmuje i zwraca parametry podane w sposób bezpośredni (niezakodowany i nieskompresowany)
2. Interfejs `DPDInfoServicesXmlEvents`, w którym parametry przekazywane są w postaci zakodowanej za pomocą base64 (opcjonalnie skompresowane w formacie zip, co ma wpływ na rozmiar przesyłanych żądań jak i odbieranych odpowiedzi)

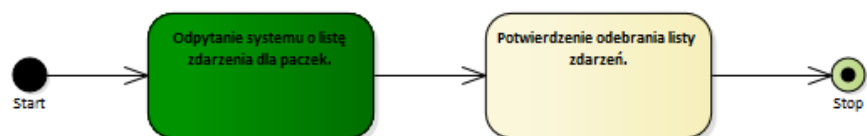
Dla większości z metod interfejsu `DPDInfoServicesXmlEvents` dostępne są dwie ich odmiany:

- zawierające w numerze wersji przedrostek „X”, które przyjmują oraz zwracają dane zakodowane w postaci base64
- zawierające w numerze wersji przedrostek „C”, które przyjmują oraz zwracają dane skompresowane za pomocą zip i jednocześnie są zakodowane w postaci base64

### 2.3.1 Interfejsy `DPDInfoServicesXmlEvents` oraz `DPDInfoServicesObjEvents`

Poniższy opis dotyczy interfejsów `DPDPackagesObjServices` oraz `DPDPackagesXMLServices`.

Opis biznesowy ich metod został przedstawiony wspólnie, ponieważ różnice pomiędzy nimi mają charakter wyłącznie strukturalny.



### 2.3.1.1 Odpytanie systemu o listę zdarzeń dla paczek z kanału klienckiego

<b>Nazwa metody</b>	getEventsForCustomer		
<b>Dostępne wersje</b>	V1 - V4	CV1 - CV4	XV1 - XV4

Metoda przyjmuje następujące parametry wejściowe:

Parametr wejściowy	Wersja metody
Maksymalna ilość zdarzeń zwrócona w odpowiedzi	1-4
Dwuliterowy kod języka opisu zdarzeń	2, 4
Dane autoryzacyjne	1-4

Po stronie systemu DPD wykonywane są następujące operacje:

Operacja	Wersja metody
Walidacja danych autoryzacyjnych	1-4
Pobranie listy zdarzeń dla paczek	1-4

Metoda zwraca następujące wartości:

Zwracana wartość	Wersja metody
Ewentualne wyjątki uwierzytelnienia i autoryzacji	1-4
Listę zdarzeń dotyczących paczek z podanego kanału klienta	1-4

Identyfikator potwierdzenia (wykorzystywany przy potwierdzeniu odebrania listy zdarzeń)	1-4
---	-----

### 2.3.1.2 Odpytanie systemu o listę zdarzeń dla paczki o podanym numerze listu przewozowego

<b>Nazwa metody</b>	getEventsForWaybill	
<b>Dostępne wersje</b>	V1	XV1

Metoda przyjmuje następujące parametry wejściowe:

<b>Parametr wejściowy</b>
Numer listu przewozowego
Dwuliterowy kod języka opisu zdarzeń
Parametr określający czy mają być zwracane wszystkie
zdarzenia ( ALL) czy tylko ostatnie ( ONLY_LAST)
Dane autoryzacyjne

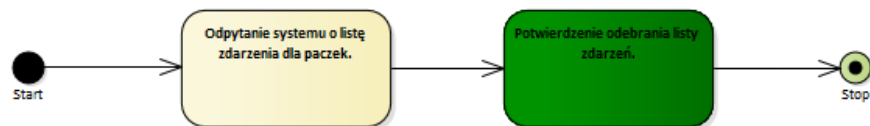
Po stronie systemu DPD wykonywane są następujące operacje:

<b>Operacja</b>
Walidacja danych autoryzacyjnych
Pobranie listy zdarzeń dla paczek

Metoda zwraca następujące wartości:

<b>Zwracana wartość</b>
Ewentualne wyjątki uwierzytelnienia i autoryzacji
Listę zdarzeń dotyczących paczki

Identyfikator potwierdzenia do wykorzystania przy potwierdzeniu odebrania
listy zdarzeń



### 2.3.1.3 Potwierdzenie odebrania listy zdarzeń

<b>Nazwa metody</b>	markEventsAsProcessed
<b>Dostępne wersje</b>	V1

Metoda przyjmuje następujące parametry wejściowe:

<b>Parametr wejściowy</b>
Identyfikator potwierdzenia
Dane autoryzacyjne

Po stronie systemu DPD wykonywane są następujące operacje:

<b>Operacja</b>
Oznaczenie listy zdarzeń odpowiadających identyfikatorowi potwierdzenia jako odebrane przez klienta

Metoda zwraca następujące wartości:

<b>Zwracana wartość</b>
Wartość "true" jeśli wywołanie zostało wykonana poprawnie, "false" w przypadku błędu

## 3 Opis struktury serwisu



Metody opisane są według następującego porządku:

- Sygnatura – postać metody po stronie webserwisu
- Parametry – opisuje wszystkie parametry występujące w strukturze żądania
- Uwagi – dodatkowy opis parametrów
- Wyniki zapytania - opis informacji zwrotnej w różnych wariantach

## 3.1 Autoryzacja

### 3.1.1 DPDInfoServicesObjEvents i DPDInfoServicesXmlEvents

Wszystkie metody z INFOServices posiadają w swej sygnaturze parametr authDataV1 (typu AuthDataV1), przechowujący dane autoryzacji do usługi. Każdy odbiorca powinien posiadać swoje własne parametry logowania. AuthDataV1 składa się z trzech elementów:

- login (String) – unikalny identyfikator przydzielany odbiorcy ze strony usługodawcy
- password (String) – hasło przyporządkowane do loginu
- channel (String) – kanał informacyjny do którego jest przypisany użytkownik

Parametr authDataV1 jest wymagany, a jego brak skutkuje komunikatem „Authorization failed to secure webserwis method”. W trakcie walidacji sprawdzane są następujące warunki jego struktury wewnętrznej:

Nazwa pola	Pole jest wymagane	Typ
login	Tak	String
password	Tak	String
channel	Tak	String

Uwierzytelnienie odbywa się po stronie WebSerwisu AuthServices. Do AuthServices jest przykazywana instancja klasy AuthenticationDataV1 zawierająca dane podane przez użytkownika. WebSerwis zwraca jako odpowiedź instancję enumeracji AuthenticationResponseStatusEnum. W przypadku, gdy autoryzacja nie powiodła się (odpowiedź jest różna od wartości „OK”), zwracany jest xml o następującej postaci:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>Access denied to secured webserwis method</faultstring>
      <detail>
        <ns2:exception
class="pl.com.dpd.wscommon.exceptions.DeniedAccessWSEException" note="To
disable this feature, set
com.sun.xml.ws.fault.SOAPFaultBuilder.disableCaptureStackTrace system
property to false" xmlns:ns2="http://jax-ws.dev.java.net/">
          <message>Access denied to secured webserwis method</message>
          <ns2:stackTrace>
            ...
          </ns2:stackTrace>
        </ns2:exception>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

```

        </ns2:exception>
    </detail>
</S:Fault>
</S:Body>
</S:Envelope>

```

## 3.2 Opis metod interfejsów

Serwis składa się z dwóch głównych interfejsów zawierających metody realizujące te same cele biznesowe oraz dwóch dodatkowych interfejsów, które służą do pobrania podpisu klienta oraz numeru wersji aplikacji. Różnica pomiędzy interfejsami `DPDInfoServicesXmlEvents` i `DPDInfoServicesObjEvents` polega na sposobach formatowania i przesyłania informacji. Poniżej znajduje się szczegółowy opis wszystkich interfejsów oraz ich metod.

### 3.2.1 Interfejs `DPDInfoServicesObjEvents`

#### 3.2.1.1 Metody związane z odpytywaniem systemu o listę zdarzeń dla paczek z kanału klienckiego

<b>Nazwa metody</b>	<code>getEventsForCustomerV1</code>
<b>Sygnatura wejścia</b>	<code>CustomerEventsResponseV1 getEventsForCustomerV1(Integer limit, AuthDataV1 authDataV1)</code>
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	<code>CustomerEventsResponseV1(String confirmId, List&lt;CustomerEventV1&gt; eventsList)</code>
<b>Kody błędów</b>	

#### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
authDataV1	AuthDataV1	Tak

#### Wynik zapytania

### Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerV1>
      <limit>10 </limit>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerV1>
  </soapenv:Body>
</soapenv:Envelope>
```

### Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V1 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        <confirmId>6/YZj7qK8rzMimm3VRYyjq==</confirmId>
        <eventsList>
          <businessCode>170101</businessCode>
          <country>PL</country>
          <depot>BIA</depot>
          <description>Wydanie przesyłki do doręczenia</description>
          <eventData1>eventData1</eventData1>
          <eventData2>eventData2</eventData2>
          <eventData3>eventData3</eventData3>
          <eventTime>2011-11-16T12:43:41</eventTime>
          <id>14123075</id>
          <objectId>46181630</objectId>
          <operationType>INSERT</operationType>
          <packageReference/>
          <parcelReference>TEL000069708012</parcelReference>
          <waybill>NG148497H090</waybill>
        </eventsList>
        <eventsList>
          <businessCode>170101</businessCode>
          <country>PL</country>
          <depot>KTW</depot>
          <description>Wydanie przesyłki do doręczenia</description>
          <eventData1>eventData1</eventData1>
          <eventData2>eventData2</eventData2>
          <eventData3>eventData3</eventData3>
          <eventTime>2011-11-16T12:48:25</eventTime>
          <id>14123116</id>
          <objectId>46183717</objectId>
          <operationType>INSERT</operationType>
          <packageReference/>
          <parcelReference>MATERIAŁY FIRMOWE</parcelReference>
          <waybill>01765003987175</waybill>
        </eventsList>
      </return>
    </ns2:getEventsForCustomerV1Response>
  </S:Body>
</S:Envelope>
```

```

        </eventsList>
    </return>
</ns2:getEventsForCustomerV1Response>
</S:Body>
</S:Envelope>

```

Prawidłowo wygenerowana odpowiedź powinna być oznaczona jako `getEventsForCustomerV1Response` i być typu `CustomerEventsResponseV1`.

### Format danych odpowiedzi:

#### CustomerEventsResponseV1:

Pole	Opis	Typ
ConfirmId	Identyfikator potwierdzenia	String
EventsList	Lista zdarzeń	List<CustomerEventV1>

#### CustomerEventV1:

Pole	Opis	Typ
Id	Identyfikator potwierdzenia	Long
BusinessCode	Kod zdarzenia	String
Waybill	Numer listu przewozowego	String
Description	Opis zdarzenia	String
EventTime	Czas wystąpienia zdarzenia	String
EventData1	Dodatkowe dane do zdarzenia	String
EventData2	Dodatkowe dane do zdarzenia	String
EventData3	Dodatkowe dane do zdarzenia	String
Depot	Odział w którym wystąpiło zdarzenie	String
Country	Kraj w którym wystąpiło zdarzenie	String
PackageReference	Kraj w którym wystąpiło zdarzenie	String
ParcelReference	Numer referencyjny paczki	String
ObjectId	Identyfikator obiektu zdarzenia	Long
OperationType	Typ operacji	String

Informacje na temat zdarzeń mogą być zwracane z różnymi typami operacji (`OperationType`):

- INSERT – nowe zdarzenie (zwracane są pełne dane dotyczące zdarzenia)
- CANCEL – anulowanie przesłanego wcześniej zdarzenia (zwracany jest jedynie identyfikator obiektu ObjectId). Zaleca się odpowiednie oznaczenie takiego zdarzenia w systemie klienckim (np. flagą „anulowano”).

### Uwagi:

Wszystkie dostępne opisy, które mogą być zwrócone w polu Description znajdują się w załączniku Zdarzenia Infoervices.xlsx.

<b>Nazwa metody</b>	<b>getEventsForCustomerV2</b>
<b>Sygnatura wejścia</b>	CustomerEventsResponseV1 getEventsForCustomerV2 (Integer limit, String language, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	CustomerEventsResponseV1( String confirmId, List<CustomerEventV1> eventsList)
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerV2>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
```

```

        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
    </authDataV1>
</even:getEventsForCustomerV2>
</soapenv:Body>
</soapenv:Envelope>

```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V2 (Response):

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerV2Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        <confirmId>ihtXUs/WAUU=</confirmId>
        <eventsList>
          <businessCode>230403</businessCode>
          <country>616</country>
          <depot>BIA</depot>
          <eventData1>parcel_3</eventData1>
          <eventData2/>
          <eventData3/>
          <eventTime>2014-11-26T11:39:39</eventTime>
          <id>20057</id>
          <objectId>0</objectId>
          <operationType>INSERT</operationType>
          <packageReference>ref</packageReference>
          <parcelReference>data29</parcelReference>
          <waybill>21112014111444</waybill>
        </eventsList>
        <eventsList>
          <businessCode>170301</businessCode>
          <country>616</country>
          <depot>BIA</depot>
          <eventData1/>
          <eventData2/>
          <eventData3/>
          <eventTime>2014-11-25T13:49:00</eventTime>
          <id>20058</id>
          <objectId>208750</objectId>
          <operationType>INSERT</operationType>
          <packageReference>ref</packageReference>
          <parcelReference>data10</parcelReference>
          <waybill>2111201411127K</waybill>
        </eventsList>
      </return>
    </ns2:getEventsForCustomerV2Response>
  </S:Body>
</S:Envelope>

```

Prawidłowo wygenerowana odpowiedź powinna być oznaczona jako `getEventsForCustomerV2Response` i być typu `CustomerEventsResponseV1`.

Format danych odpowiedzi jest analogiczny jak w przypadku metody `getEventsForCustomerV1`.

<b>Nazwa metody</b>	<b>getEventsForCustomerV3</b>
<b>Sygnatura wejścia</b>	CustomerEventsResponseV2 getEventsForCustomerV3( Integer limit, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	CustomerEventsResponseV2 ( String confirmId, List<CustomerEventV2> eventsList)
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerV3>
      <limit>10</limit>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerV3>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V3 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerV3Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
```

```

<confirmId>ihtXUs/WAUU=</confirmId>
<eventsList>
  <businessCode>230403</businessCode>
  <country>616</country>
  <depot>BIA</depot>
  <eventDataList>
    <code>01</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventDataList>
    <code>02</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventDataList>
    <code>03</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventTime>2014-11-26T11:39:39</eventTime>
  <id>20057</id>
  <objectId>0</objectId>
  <operationType>INSERT</operationType>
  <packageReference>ref</packageReference>
  <parcelReference>data29</parcelReference>
  <waybill>21112014111444</waybill>
</eventsList>
<eventsList>
  <businessCode>170301</businessCode>
  <country>616</country>
  <depot>BIA</depot>
  <eventDataList>
    <code>02</code>
    <value>parcel_1</value>
  </eventDataList>
  <eventDataList>
    <code>03</code>
    <value>parcel_1</value>
  </eventDataList>
  <eventTime>2014-11-25T13:49:00</eventTime>
  <id>20058</id>
  <objectId>208750</objectId>
  <operationType>INSERT</operationType>
  <packageReference>ref</packageReference>
  <parcelReference>data10</parcelReference>
  <waybill>2111201411127K</waybill>
</eventsList>
</return>
</ns2:getEventsForCustomerV3Response>
</S:Body>
</S:Envelope>

```

Prawidłowo wygenerowana odpowiedź powinna być oznaczona jako getEventsForCustomerV3Response i być typu CustomerEventsResponseV2.

### Format danych odpowiedzi



**CustomerEventsResponseV1:**

Pole	Opis	Typ
ConfirmId	Identyfikator potwierdzenia	String
EventsList	Lista zdarzeń	List<CustomerEventV2>

**CustomerEventV2:**

Pole	Opis	Typ
Id	Identyfikator potwierdzenia	Long
BusinessCode	Kod zdarzenia	String
Waybill	Numer listu przewozowego	String
Description	Opis zdarzenia	String
EventTime	Czas wystąpienia zdarzenia	String
Depot	Oddział w którym wystąpiło zdarzenie	String
Country	Kraj w którym wystąpiło zdarzenie	String
PackageReference	Kraj w którym wystąpiło zdarzenie	String
ParcelReference	Numer referencyjny paczki	String
ObjectId	Identyfikator obiektu zdarzenia	Long
EventDataList	Dodatkowe dane do zdarzenia	List<CustomerEventDataV2>

**CustomerEventDataV2:**

Pole	Opis	Typ
Code	Kod danych	String
Value	Value	String

Nazwa metody	getEventsForCustomerV4
Sygnatura wejścia	CustomerEventsResponseV2 getEventsForCustomerV4 ( Integer limit, String language, AuthDataV1 authDataV1)

<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	CustomerEventsResponseV2 ( String confirmId, List<CustomerEventV2> eventsList)
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerV4>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerV4>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V3 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerV4Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        <confirmId>ihtXUs/WAUU=</confirmId>
      </return>
    </ns2:getEventsForCustomerV4Response>
  </S:Body>
</S:Envelope>
```

```

<eventsList>
  <businessCode>230403</businessCode>
  <country>616</country>
  <depot>BIA</depot>
  <eventDataList>
    <code>01</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventDataList>
    <code>02</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventDataList>
    <code>03</code>
    <value>&lt;![CDATA[parcel_3]]&gt;</value>
  </eventDataList>
  <eventTime>2014-11-26T11:39:39</eventTime>
  <id>20057</id>
  <objectId>0</objectId>
  <operationType>INSERT</operationType>
  <packageReference>ref</packageReference>
  <parcelReference>data29</parcelReference>
  <waybill>21112014111444</waybill>
</eventsList>
<eventsList>
  <businessCode>170301</businessCode>
  <country>616</country>
  <depot>BIA</depot>
  <eventDataList>
    <code>02</code>
    <value>parcel_1</value>
  </eventDataList>
  <eventDataList>
    <code>03</code>
    <value>parcel_1</value>
  </eventDataList>
  <eventTime>2014-11-25T13:49:00</eventTime>
  <id>20058</id>
  <objectId>208750</objectId>
  <operationType>INSERT</operationType>
  <packageReference>ref</packageReference>
  <parcelReference>data10</parcelReference>
  <waybill>2111201411127K</waybill>
</eventsList>
</return>
</ns2:getEventsForCustomerV4Response>
</S:Body>
</S:Envelope>

```

Prawidłowo wygenerowana odpowiedź powinna być oznaczona jako `getEventsForCustomerV4Response` i być typu `CustomerEventsResponseV2`.

Format danych odpowiedzi jest analogiczny jak w przypadku metody `getEventsForCustomerV3`.

### 3.2.1.2 Odpytanie systemu o listę zdarzeń dla paczki o podanym numerze listu przewozowego

<b>Nazwa metody</b>	<b>getEventsForWaybillV1</b>
<b>Sygnatura wejścia</b>	CustomerEventsResponseV3 getEventsForWaybillV1( String waybill, EventsSelectTypeEnum eventsSelectType, String language, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>waybill</b> – numer listu przewozowego <b>eventsSelectType</b> – określa czy mają być zwracane wszystkie zdarzenia czy tylko ostatnie <b>authDataV1</b> – dane autoryzacyjne. Format AuthDataV1 <b>language</b> - język opisu zdarzenia
<b>Wyjście</b>	CustomerEventsResponseV3( String confirmId, List<CustomerEventV3> eventsList)
<b>Kody błędów</b>	

#### Parametry wejściowe

<b>Pole</b>	<b>Typ</b>	<b>Pole jest wymagane</b>	<b>Dodatkowe informacje</b>
waybill	Integer	Tak	
eventsSelectType	EventsSelectTypeEnum	Tak	
language	String	Nie	PL lub EN (jeśli podana zostanie inna wartość przyjmowane jest EN)
authDataV1	AuthDataV1	Tak	

**eventsSelectType** - enumeracja określająca czy mają być zwracane wszystkie zdarzenia czy tylko ostatnie:

- ALL - wszystkie zdarzenia
- ONLY\_LAST - ostatnie zdarzenia

## Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForWaybillV1>
      <waybill>sampleWaybill</waybill>
      <eventsSelectType>ONLY_LAST</eventsSelectType>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForWaybillV1>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V1 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForWaybillV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        <confirmId>0</confirmId>
        <eventsList>
          <businessCode>190101</businessCode>
          <country>PL</country>
          <depot>1305</depot>
          <depotName>Warszawa</depotName>
          <description>Przesyłka doręczona</description>
          <eventDataList>
            <code>03</code>
            <description>Odbiorca</description>
            <value>test</value>
          </eventDataList>
          <eventTime>2018-02-15T15:52:29.886</eventTime>
          <objectId>7305000000269747964</objectId>
          <packageReference/>
          <parcelReference/>
          <waybill>0000001094090S</waybill>
        </eventsList>
      </return>
    </ns2:getEventsForWaybillV1Response>
  </S:Body>
</S:Envelope>
```

Prawidłowo wygenerowana odpowiedź powinna być oznaczona jako getEventsForWaybillV1Response i być typu CustomerEventsResponseV2.

Struktura odpowiedzi XML dla nieprawidłowego zapytania w wersji V1, czyli takiego, w którym waybill nie został uzupełniony przez użytkownika:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForWaybillV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        <confirmId>0</confirmId>
      </return>
    </ns2:getEventsForWaybillV1Response>
  </S:Body>
</S:Envelope>
```

### Format danych odpowiedzi

#### CustomerEventsResponseV3:

Pole	Opis	Typ
ConfirmId	Identyfikator potwierdzenia	String
EventsList	Lista zdarzeń	List<CustomerEventV3>

#### CustomerEventV3:

Pole	Opis	Typ
BusinessCode	Kod zdarzenia	String
Waybill	Numer listu przewozowego	String
Description	Opis zdarzenia	String
EventTime	Czas wystąpienia zdarzenia	String
Depot	Odział w którym wystąpiło zdarzenie	String
Country	Kraj w którym wystąpiło zdarzenie	String
PackageReference	Kraj w którym wystąpiło zdarzenie	String
ParcelReference	Numer referencyjny paczki	String
ObjectId	Identyfikator obiektu zdarzenia	Long
EventDataList	Dodatkowe dane do zdarzenia	List<CustomerEventDataV3>

#### CustomerEventDataV3:

Pole	Opis	Typ
Code	Kod danych	String
Value	Value	String
Description	Opis danej	String

### 3.2.1.3 Potwierdzenie odebrania listy zdarzeń

<b>Nazwa metody</b>	<b>markEventsAsProcessedV1</b>
<b>Sygnatura wejścia</b>	Boolean markEventsAsProcessedV1 ( String confirmId, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>confirmId</b> – identyfikator potwierdzenia <b>authDataV1</b> – Dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Typ logiczny o wartości "true" przy pomyślnym wykonaniu metody, "false" gdy wykonanie metody nie powiedzie się
<b>Kody błędów</b>	

#### Parametry wejściowe

Pole	Typ	Pole jest wymagane
confirmId	String	Tak

#### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:markEventsAsProcessedV1>
      <confirmId>sampleConfirmId</confirmId>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:markEventsAsProcessedV1>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    </soapenv:Body>
</soapenv:Envelope>

```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji V1 (Response):

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:markEventsAsProcessedV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>true</return>
    </ns2:markEventsAsProcessedV1Response>
  </S:Body>
</S:Envelope>

```

Struktura odpowiedzi XML dla nieprawidłowego zapytania w wersji V1, czyli takiego, w którym confirmId nie został uzupełniony lub jest błędny:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:markEventsAsProcessedV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>false</return>
    </ns2:markEventsAsProcessedV1Response>
  </S:Body>
</S:Envelope>

```

### 3.2.2 Interfejs DPDInfoServicesXmlEvents

#### 3.2.2.1 Metody związane z odpytaniem systemu o listę zdarzeń dla paczek z kanału klienckiego

<b>Nazwa metody</b>	getEventsForCustomerCV1
<b>Sygnatura wejścia</b>	byte[] getEventsForCustomerCV1( Integer limit, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu CustomerEventsResponseV1) skompresowany metodą ZIP i zakodowany za pomocą base64
<b>Kody błędów</b>	

#### Parametry wejściowe

Pole	Typ	Pole jest wymagane
------	-----	--------------------



confirmId	String	Tak
-----------	--------	-----

## Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerCV1>
      <limit>10 </limit>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerCV1>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji CV1 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerCV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
        UEsDBBQACAAIAJKEVEwAAAAAAAAAAAAAAAAAAAAWAAAAQ3VzdG9tZXJFdmVudHNSZXNwb25zZdWTX0/CM
        BTFPxFpbzsYkGsT/j0QjRgc4psp46JV2Ja2mPDt7TJ1Q4X44ItJk7bnnNs295fiaO98viM7eaXMuz
        m5Is8c3YHCUZ5tjn1N18o8+fuFY8vBYnGBrNaxKroyzod486CyPgQE5+0YWRkd7p3JyLlRviYlJI+
        4RHYk4lIfVma7VQIABICoTFEUIfvQq+sSwsHBLsF0BKdBKave2Egq+0qOdZegyq0TWn7IN/9SqwD
        QjUM0TBk0wibMRW5V8PpAFm1DA3aZ94eVAc6ZVeQDd7o9EU/0pw2ZC1LSVnaIPumhlz5r1pY1/f3y
        uSxjrPVM6U+9JAj+1zjrCCrvcmz5FCQml7fTuZJ8I/U8KqvUE5R6v5ACWIuOfyCkogvz1NqJyD7Ua
        /P+S1K/4EO8HN0BO/G7b9AxJq/ip38n29QSwcIvhp3Ez8BAAC6AwAAUESBAHQAFAAIAAgAkoRUTL4
        adxM/AQAAugMAABYAAAAAAAAAAAAAAAAAAAAAEN1c3RvbWVyRXZlbnRzUmVzcG9uc2VQSwUGAAAA
        AAEAAQBEAAAAgwEAAAA
      </return>
    </ns2:getEventsForCustomerCV1Response>
  </S:Body>
</S:Envelope>
```

**Format danych odpowiedzi zdekompresowanej i zdekodowanej z postaci base64 jest analogiczny jak w przypadku metody getEventsForCustomerV1.**

Przykładowe archiwum zawierające odpowiedź zdekodowaną z postaci base64 znajduje się w załączniku DECODED\_CV1\_RESPONSE.zip

Nazwa metody	getEventsForCustomerCV2
--------------	-------------------------

<b>Sygnatura wejścia</b>	byte[] getEventsForCustomerCV2 ( Integer limit, String language, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu CustomerEventsResponseV1) skompresowany metodą ZIP i zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerCV2>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerCV2>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji CV2 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
```

```

<ns2:getEventsForCustomerCV2Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
  <return>
UESDBBQACAAIAEmFVEwAAAAAAAAAAAAAAAAAAAAWAAAAQ3VzdG9tZXJFdmVudHNSZXNwb25zZdWTX0/CM
BTFPxFpbzsYkGsT/j0QjRgc4psp46JV2Ja2mPDt7TJ1Q4X44ItJk7bnnNs295fiaO98viM7eaXMuz
m5Is8c3YHCUZ5tjN1N18o8+fuFY8vBYnGBrNaxKroyzod486CyPgQE5+0YWRkd7p3JyLlRviYlJI+
4RHYk4lIfVma7VQIABICoTFEUIfvQq+sSwsHBLsF0BKdBKAVE2Egq+0qOdZegyq0TWn7IN/9SqWd
QjUM0TBk0wibMRW5V8PpAFm1DA3aZ94eVAc6ZVeQDd7o9EU/0pw2ZC1LSVnaIPumhlz5rlpY1/f3y
uSxjrPVM6U+9JAj+1zjrCCrvczm5FCQml7fTuZJ8I/U8KqvUE5R6v5ACWIuOfyCkogvz1NqJyD7Ua
/P+S1K/4EO8HN0BO/G7b9AxJq/ip38n29QSwcIvhp3Ez8BAAC6AwAAUESBAHQAFAAIAAgASYVUTL4
adxM/AQAAugMAABYAAAAAAAAAAAAAAAAAAAAAEN1c3RvbWVYRXZlbnRzUmVzcG9uc2VQSwUGAAAA
AAEAQAQBEAAAAgweAAAAA
  </return>
</ns2:getEventsForCustomerCV2Response>
</S:Body>
</S:Envelope>

```

**Format danych odpowiedzi zdekompresowanej i zdekodowanej z postaci base64** jest analogiczny jak w przypadku metody `getEventsForCustomerV1`.

Przykładowe archiwum zawierające odpowiedź zdekodowaną z postaci base64 znajduje się w załączniku `DECODED_CV1_RESPONSE.zip` (jest ono takie same jak w przypadku `getEventsForCustomerCV1`).

<b>Nazwa metody</b>	<code>getEventsForCustomerCV3</code>
<b>Sygnatura wejścia</b>	<code>byte[] getEventsForCustomerCV3( Integer limit, AuthDataV1 authDataV1)</code>
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu <code>CustomerEventsResponseV2</code> ) skompresowany metodą ZIP i zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
authDataV1	AuthDataV1	Tak

## Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerCV3>
      <limit>10</limit>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerCV3>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji CV3 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerCV3Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
UESDBBQACAAIAI2FVEwAAAAAAAAAAAAAAAAAAAAWAAAAQ3VzdG9tZXJFdmVudHNSZXNwb25zZb1U207CQ
BT8FeWBN9xLy/2whJsJ0ajBAiaEmAIHrJa26S5E/t4tFZEC4gOabNLdOZMzPbOTHer73L1YYigd36
uk2BVNXaA39ieON6ukutZ1ppCqCmgspPLnGLaW6CnZQRn4nsQeFxAjt45UCVZUbE8EpzSbB6J3UF9
Ix0MpG/4EBTeoSQ0gOyD07dXIcV3BGWOCm1N/TNMEssFjOcuZ6wa6nGEsw3MWYyWjqBeQbRmaGPhK
1Ns1IPEWgv7CU+FK5FgOyOYAD/b4zz5hB6cY6s1RhDgFsodqXjhGdwtMbGXzYsTcxeE+wNBW2k5rF
aBo3z22OhaQXTSeo61bHHAugiP31pZQFv1r5E3Pdhco0q4qXw4azZpVGWRr6WdjOEzPVB1IzND8g9
2OS/C/1zDOJEESxpG9zB0LYeFACFmeGpHBj0PI8zc/hzBrMaNkFkuU/nsIGT0UwtErjtV69EI+qx1
fwLnzmQjP522ys0X1lw1PB4PsPFW+N3XCufbDeVFPXUn6tW63Eilv8ESH74/eB1BLBwhZCbhTjwEA
ADUFAABQSwECFAAUAAgACACNhrVMWQm4U48BAAA1BQAAFgAAAAAAAAAAAAAAAAAAAAAAQ3VzdG9tZ
XJFdmVudHNSZXNwb25zZVBLBQYAAAAAAQABAEQAAADTAQAAAAA
      </return>
    </ns2:getEventsForCustomerCV3Response>
  </S:Body>
</S:Envelope>
```

**Format danych odpowiedzi zdekompresowanej i zdekodowanej z base64** jest analogiczny jak w przypadku metody getEventsForCustomerV3.

Przykładowe archiwum zawierające odpowiedź zdekodowaną z postaci base64 znajduje się w załączniku DECODED\_CV3\_RESPONSE.zip.

Nazwa metody	getEventsForCustomerCV4
--------------	-------------------------

<b>Sygnatura wejścia</b>	byte[] getEventsForCustomerCV4 ( Integer limit, String language, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu CustomerEventsResponseV2) skompresowany metodą ZIP i zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerCV4>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerCV4>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji CV4 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<S:Body>
  <ns2:getEventsForCustomerCV4Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
    <return>
UESDBBQACAAIALmBVEwAAAAAAAAAAAAAAAAAAAAAAQ3VzdG9tZXJFdmVudHNSZXNwb25zZbOxr8jNU
ShLLSrOzM+zVTLUM1BSSM1Lzk/JzEu3VQoNcd01ULK3s3EuLS7Jz00tcilLzSspDkotLsjPK04NM7
KzgYj4ZBaX6AOV5eelZRbleqbYBZS5JxbmJFVqO2fb2ugjxIFsXEYBAFBLBwiok5GDaQAAAIsAAAB
QSwECFAAUAAgACAC5gVRMqJORg2kAAACLAAAAFgAAAAAAAAAAAAAAAAAAAAAAQ3VzdG9tZXJFdmVu
dHNSZXNwb25zZVBLBQYAAAAAAQABAEQAAACtAAAAAA=
    </return>
  </ns2:getEventsForCustomerCV4Response>
</S:Body>
</S:Envelope>

```

**Format danych odpowiedzi zdekompresowanej i zdekodowanej z base64** jest analogiczny jak w przypadku metody `getEventsForCustomerV3`.

Przykładowe archiwum zawierające odpowiedź zdekodowaną z postaci base64 znajduje się w załączniku `DECODED_CV3_RESPONSE.zip` (jest takie same jak w przypadku `getEventsForCustomerCV3`).

<b>Nazwa metody</b>	<code>getEventsForCustomerXV1</code>
<b>Sygnatura wejścia</b>	<code>byte[] getEventsForCustomerXV1( Integer limit, AuthDataV1 authDataV1)</code>
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu <code>CustomerEventsResponseV1</code> zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
<code>confirmId</code>	String	Tak
<code>authDataV1</code>	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerXV1>
      <limit>10</limit>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerXV1>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji XV1 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerXV1Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
PEN1c3RvbWVyRXZ1bnRzUmVzcG9uc2VWMT48Q229uZmlybUlkPmlodFhVcy9XQVVPPTwvQ229uZmlybUlkPjxFdmVudHNMaXN0PjxDdXN0b211ckV2ZW50VjE+PElkPjIwMDU3PC9JZD48QnVzaW51c3NDb2RlPjIzMDQwMzwwQnVzaW51c3NDb2RlPjxXYXliaWxsPjIxMTEyMDE0MTEeNDQ0PC9XYXliaWxsPjxFdmVudFRpbWU+MjAxNC0xMS0yNlQxMT0zOT0zOTwvRXZ1bnRUaW1lPjxFdmVudERhdGEePnBhcmN1bF8zPC9FdmVudERhdGEePjxFdmVudERhdGEyPjwvRXZ1bnREYXRhMj48RXZ1bnREYXRhMz48L0V2ZW50RGF0YTM+PERlcG90PkJJQTwwRGVwb3Q+PENvdW50cnk+NjE2PC9Db3VudHJ5PjxQYWNRyWdlUmVmZXJlbnNlPnJlZjwvUGFja2FnZVJlZmV5ZW5jZT48UGFyY2VsUmVmZXJlbnNlPmRhdGEyOTwvUGFyY2VsUmVmZXJlbnNlPjxPYmplY3RJZD4wPC9PYmplY3RJZD48T3BlcmF0aW9uVHlwZT5JTlNFU1Q8L09wZXJhdGlvb1R5cGU+PC9DdXN0b211ckV2ZW50VjE+PEN1c3RvbWVyRXZ1bnRWMT48SWQ+MjAwNTg8L0lkPjxCdXNpbmVzc0NvZGU+MTcwMzAxPC9CdXNpbmVzc0NvZGU+PFdheWJpbGw+MjExMTIwMTQxMTEyN0s8L1dheWJpbGw+PEV2ZW50VGltZT4yMDE0LTEXLTI1VDEzOjQ5OjAwPC9FdmVudFRpbWU+PEV2ZW50RGF0YTE+PC9FdmVudERhdGEePjxFdmVudERhdGEyPjwvRXZ1bnREYXRhMj48RXZ1bnREYXRhMz48L0V2ZW50RGF0YTM+PERlcG90PkJJQTwwRGVwb3Q+PENvdW50cnk+NjE2PC9Db3VudHJ5PjxQYWNRyWdlUmVmZXJlbnNlPnJlZjwvUGFja2FnZVJlZmV5ZW5jZT48UGFyY2VsUmVmZXJlbnNlPmRhdGEeMDwvUGFyY2VsUmVmZXJlbnNlPjxPYmplY3RJZD4yMDg3NTA8L09iamVjdElkPjxPcGVyYXRpb25UeXB1Pk1OU0VSVDwvT3BlcmF0aW9uVHlwZT48L0N1c3RvbWVyRXZ1bnRWMT48L0V2ZW50c0xp c3Q+PC9DdXN0b211ckV2ZW50c1Jlc3BvbmlvJE+
      </return>
    </ns2:getEventsForCustomerXV1Response>
  </S:Body>
</S:Envelope>
```

Format danych odpowiedzi dekodowanej z base64 jest analogiczny jak w przypadku metody getEventsForCustomerV1.

Nazwa metody	getEventsForCustomerXV2
Sygnatura wejścia	byte[] getEventsForCustomerXV2 ( Integer limit, String language, AuthDataV1 authDataV1)

<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu CustomerEventsResponseV1) zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerXV2>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerXV2>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji XV2 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForCustomerXV2Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
```



```

PEN1c3RvbWVyRXZ1bnRzUmVzcG9uc2VWMT48Q29uZmlybUlkPmlodFhVcy9XQVVPVPTwvQ29uZmlybUlkPjxFdmVudHNMaXN0PjxDdXN0b211ckV2ZW50VjE+PElkPjIwMDU3PC9JZD48QnVzaW51c3NDb2RlPjIzMDQwMzwvQnVzaW51c3NDb2RlPjxXYXliaWxsPjIxMTEyMDE0MTEeNDQ0PC9XYXliaWxsPjxFdmVudFRpbWU+MjAxNC0xMS0yNlQxMT0zOT0zOTwvRXZ1bnRUaW1lPjxFdmVudERhdGEePnBhcmNlbF8zPC9FdmVudERhdGEePjxFdmVudERhdGEyPjwvRXZ1bnREYXRhMj48RXZ1bnREYXRhMz48L0V2ZW50RGF0YTM+PERlcG90PkJJQTwwRGVwb3Q+PENvdW50cnk+NjE2PC9Db3VudHJ5PjxQYWNrYWdlUmVmZXJlbmNlPnJlZjwvUGFja2FnZVJlZmV5ZW5jZT48UGFyY2VsUmVmZXJlbmNlPmRhdGEyOTwvUGFyY2VsUmVmZXJlbmNlPjxPYmplY3RJZD4wPC9PYmplY3RJZD48T3B1cmF0aW9uVHlwZT5JT1NFU1Q8L09wZXJhdGlvb1R5cGU+PC9DdXN0b211ckV2ZW50VjE+PEN1c3RvbWVyRXZ1bnRWMT48SWQ+MjAwNTg8L0lkPjxCdXNpbmVzc0NvZGU+MTcwMzAxPC9CdXNpbmVzc0NvZGU+PFdheWJpbGw+MjExMTIwMTQxMTEyN0s8L1dheWJpbGw+PEV2ZW50VG1tZT4yMDE0LTExLTl1VDEzOjQ5OjAwPC9FdmVudFRpbWU+PEV2ZW50RGF0YTE+PC9FdmVudERhdGEePjxFdmVudERhdGEyPjwvRXZ1bnREYXRhMj48RXZ1bnREYXRhMz48L0V2ZW50RGF0YTM+PERlcG90PkJJQTwwRGVwb3Q+PENvdW50cnk+NjE2PC9Db3VudHJ5PjxQYWNrYWdlUmVmZXJlbmNlPnJlZjwvUGFja2FnZVJlZmV5ZW5jZT48UGFyY2VsUmVmZXJlbmNlPmRhdGEeMDwvUGFyY2VsUmVmZXJlbmNlPjxPYmplY3RJZD4yMDg3NTA8L09iamVjZElkPjxPcGVyYXRpb25UeXB1Pk1OU0VSVDwvT3B1cmF0aW9uVHlwZT48L0N1c3RvbWVyRXZ1bnRWMT48L0V2ZW50c0xpY3Q+PC9DdXN0b211ckV2ZW50c1Jlc3BvbmlvPjE+
</return>
</ns2:getEventsForCustomerXV2Response>
</S:Body>
</S:Envelope>

```

Format danych odpowiedzi dekodowanej z base64 jest analogiczny jak w przypadku metody `getEventsForCustomerV1`.

Nazwa metody	<code>getEventsForCustomerXV3</code>
Sygnatura wejścia	<code>byte[] getEventsForCustomerXV3(Integer limit, AuthDataV1 authDataV1)</code>
Parametry wejściowe	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>authDataV1</b> – dane używane do autoryzacji użytkownika
Wyjście	Obiekt w formacie XML (typu <code>CustomerEventsResponseV2</code> ) zakodowany za pomocą base64
Kody błędów	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
<code>limit</code>	<code>Integer</code>	Nie
<code>authDataV1</code>	<code>AuthDataV1</code>	Tak

### Wynik zapytania

Struktura zapytania XML (Request):



<b>Sygnatura wejścia</b>	byte[] getEventsForCustomerXV4 ( Integer limit, String language, AuthDataV1 authDataV1)
<b>Parametry wejściowe</b>	<b>limit</b> – maksymalna ilość zdarzeń zwrócona w odpowiedzi <b>language</b> – dwuliterowy kod języka opisu zdarzeń <b>authDataV1</b> – dane używane do autoryzacji użytkownika
<b>Wyjście</b>	Obiekt w formacie XML (typu CustomerEventsResponseV2) zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane
limit	Integer	Nie
language	String	Nie
authDataV1	AuthDataV1	Tak

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForCustomerV4>
      <limit>10</limit>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
        <password>userPassword</password>
      </authDataV1>
    </even:getEventsForCustomerV4>
  </soapenv:Body>
</soapenv:Envelope>
```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji XV4 (Response):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<S:Body>
  <ns2:getEventsForCustomerXV4Response
xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
    <return>
PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48Q3VzdG9tZXJFdmVudHNSZXNwb
25zZVYyPjxmdmVudHNMxN0PjxDdXN0b211ckV2ZW50VjI+PElkPjIwMDU3PC9JZD48QnVzaW5lc3
NDb2RlPjIzMDQwMzwwQnVzaW5lc3NDb2RlPjxXYXliaWxsPjIxMTEyMDE0MTExNDQ0PC9XYXliaWx
sPjxmdmVudFRpbWU+MjAxNC0xMS0yNlQxMT0zOT0zOTwvRXZlbnRUaW1lPjxEZXBvdD5CSUE8L0Rl
cG90PjxDb3VudHJ5PjYxNjwvQ291bnRyeT48UGFja2FnZVJlZmV5ZW5jZT5yZWY8L1BhY2thZ2VSZ
WZlcmVuY2U+PFBhcmNlbFJlZmV5ZW5jZT5kYXRhMjk8L1BhcmNlbFJlZmV5ZW5jZT48T3B1cmF0aW
9uVHlwZT5JTlNFU1Q8L09wZXJhdGlvb1R5cGU+PEV2ZW50RGF0YUxpc3Q+PEN1c3RvbWVvRXZlbnR
EYXRhVjI+PENvZGU+MDE8L0NvZGU+PFZhbHVlPiZsdDshW0NEQVRBW3BhcmNlbF8zXV0mZ3Q7PC9W
YWx1ZT48L0N1c3RvbWVvRXZlbnREYXRhVjI+PEN1c3RvbWVvRXZlbnREYXRhVjI+PENvZGU+MDI8L
0NvZGU+PFZhbHVlPiZsdDshW0NEQVRBW3BhcmNlbF8zXV0mZ3Q7PC9WYWx1ZT48L0N1c3RvbWVvRX
ZlbnREYXRhVjI+PEN1c3RvbWVvRXZlbnREYXRhVjI+PENvZGU+MDM8L0NvZGU+PFZhbHVlPiZsdDs
hW0NEQVRBW3BhcmNlbF8zXV0mZ3Q7PC9WYWx1ZT48L0N1c3RvbWVvRXZlbnREYXRhVjI+PC9FdmVu
dERhdGFMaXN0PjwvQ3VzdG9tZXJFdmVudFYyPjxDdXN0b211ckV2ZW50VjI+PElkPjIwMDU4PC9JZ
D48QnVzaW5lc3NDb2RlPjE3MDMwMTwvQnVzaW5lc3NDb2RlPjxXYXliaWxsPjIxMTEyMDE0MTExNDQ0
PC9XYXliaWxsPjxmdmVudFRpbWU+MjAxNC0xMS0yNVQxMz00OTowMDwvRXZlbnRUaW1lPjxEZXB
vdD5CSUE8L0RlcG90PjxDb3VudHJ5PjYxNjwvQ291bnRyeT48UGFja2FnZVJlZmV5ZW5jZT5yZWY8
L1BhY2thZ2VSZWZlcmVuY2U+PFBhcmNlbFJlZmV5ZW5jZT5kYXRhMTA8L1BhcmNlbFJlZmV5ZW5jZ
T48T2JqZW50SWQ+MjA4NzUwPC9PYmplY3RjZD48T3B1cmF0aW9uVHlwZT5JTlNFU1Q8L09wZXJhdG
lvb1R5cGU+PEV2ZW50RGF0YUxpc3Q+PEN1c3RvbWVvRXZlbnREYXRhVjI+PENvZGU+MDI8L0NvZGU
+PFZhbHVlPnBhcmNlbF8xPC9WYWx1ZT48L0N1c3RvbWVvRXZlbnREYXRhVjI+PEN1c3RvbWVvRXZl
bnREYXRhVjI+PENvZGU+MDM8L0NvZGU+PFZhbHVlPnBhcmNlbF8xPC9WYWx1ZT48L0N1c3RvbWVvR
XZlbnREYXRhVjI+PC9FdmVudERhdGFMaXN0PjwvQ3VzdG9tZXJFdmVudFYyPjwvRXZlbnRzTG1zdD
48Q29uZmlybUlkPmlodFhVcy9XQVVPVPTwvQ29uZmlybUlkPjwvQ3VzdG9tZXJFdmVudHNSZXNwb25
zZVYyPg==
    </return>
  </ns2:getEventsForCustomerXV4Response>
</S:Body>
</S:Envelope>

```

Format danych odpowiedzi dekodowanej z base64 jest analogiczny jak w przypadku metody getEventsForCustomerV3.

### 3.2.2.2 Odpytanie systemu o listę zdarzeń dla paczki o podanym numerze listu przewozowego

Nazwa metody	getEventsForWaybillXV1
Sygnatura wejścia	byte[] getEventsForWaybillXV1( String waybill, EventsSelectTypeEnum eventsSelectType, String language, AuthDataV1 authDataV1)

<b>Parametry wejściowe</b>	<b>waybill</b> – numer listu przewozowego <b>eventsSelectType</b> – określenie, czy mają być zwracane wszystkie zdarzenia, czy tylko ostatnie <b>language</b> – język opisu zdarzenia <b>authDataV1</b> – dane autoryzacyjne. Format AuthDataV1
<b>Wyjście</b>	Obiekt w formacie XML (typu <b>CustomerEventResponseV3</b> ) zakodowany za pomocą base64
<b>Kody błędów</b>	

### Parametry wejściowe

Pole	Typ	Pole jest wymagane	Dodatkowe informacje
waybill	Integer	Tak	
eventsSelectType	EventsSelectTypeEnum	Tak	
language	String	Nie	PL lub EN (jeśli podana zostanie inna wartość przyjmowane jest EN)
authDataV1	AuthDataV1	Tak	

**eventsSelectType** - enumeracja określająca czy mają być zwracane wszystkie zdarzenia czy tylko ostatnie:

- ALL - wszystkie zdarzenia
- ONLY\_LAST - ostatnie zdarzenia

### Wynik zapytania

Struktura zapytania XML (Request):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://events.dpdinfoservices.dpd.com.pl/">
  <soapenv:Header/>
  <soapenv:Body>
    <even:getEventsForWaybillXV1>
      <waybill>sampleWaybill</waybill>
      <eventsSelectType>ONLY_LAST</eventsSelectType>
      <language>PL</language>
      <authDataV1>
        <channel>clientChannel</channel>
        <login>user</login>
      </authDataV1>
    </even:getEventsForWaybillXV1>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <password>userPassword</password>
    </authDataV1>
</even:getEventsForWaybillXV1>
</soapenv:Body>
</soapenv:Envelope>

```

Struktura odpowiedzi XML dla prawidłowego zapytania w wersji XV1 (Response):

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getEventsForWaybillXV1Response
      xmlns:ns2="http://events.dpdinfoservices.dpd.com.pl/">
      <return>
PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48Q3VzdG9tZXJFdmVudHNSZXNwb
25zZVYzPjxFdmVudHNMaXN0PjxDdXN0b211ckV2ZW50VjM+PEJlc2luZXNzQ29kZT4xNzAxMDI8L0
Jlc2luZXNzQ29kZT48V2F5YmlsbD4xMzA1OTkwMDAwMDA2NDwvV2F5YmlsbD48RGVzY3JpcHRpb24
+V3lkYW5pZSBwcnplc3NfgmtpIGRvIGRvczSY3plbmhpc9EZjcmldG1vbj48RXZlbnRUaW1l
PjIwMTgtMDItMjZUMTE6MTE6NTcuNjQyPC9FdmVudFRpbWU+PERlcG90PjEzMDU8L0RlcG90PjxEZ
XBvdE5hbWU+V2Fyc3phd2E8L0RlcG90TmFtZT48Q291bnRyeT5QTdWvQ291bnRyeT48UGFja2FnZV
JlZmVyZW5jZT5SRUYvMS8wMDI5Mi8yMDE3PC9QYWNrYWdlUmVmZXJlbnNlPjxQYXJjZWxSZWZlcmV
uY2U+UkVGIzAwMDAwMDAwMDAwMTIzPC9QYXJjZWxSZWZlcmVuY2U+PE9iamVjdElkPjczMDUw
MDAwMDAyNzU2MjMwNDk8L09iamVjdElkPjxFdmVudERhdGFMaXN0Lz48L0Nlc3RvbWVvRXZlbnRWM
z48L0V2ZW50c0xpc3Q+PENvbmZpcmlJZD4wPC9Db25maXJtSWQ+PC9DdXN0b211ckV2ZW50c1Jlc3
BvbnNlVjM+
      </return>
    </ns2:getEventsForWaybillXV1Response>
  </S:Body>
</S:Envelope>

```

Format danych odpowiedzi dekodowanej z base64 jest analogiczny jak w przypadku metody `getEventsForWaybillV1`.

### 3.2.2.3 Potwierdzenie odebrania listy zdarzeń

Metoda `markEventsAsProcessedV1` jest identyczna z metodą opisaną w ramach opisu interfejsu `DPDInfoServicesObjEvents`.

## 3.3 Kody błędów

Brak

## 4 Przekierowania i zwroty

Numer listów przewozowych / przesyłek mogą się czasami zmieniać i w takich przypadkach należy śledzić nowy numer przesyłki. Może się to zdarzyć w następujących sytuacjach:

- 230402 - przekierowanie przesyłki
- 230403, 230408 - przesyłka do zwrotu do nadawcy

Nowy numer śledzenia jest zwracany w elemencie `<eventDataList><value>` w odpowiedziach ze zdarzeniami wymienionymi powyżej.