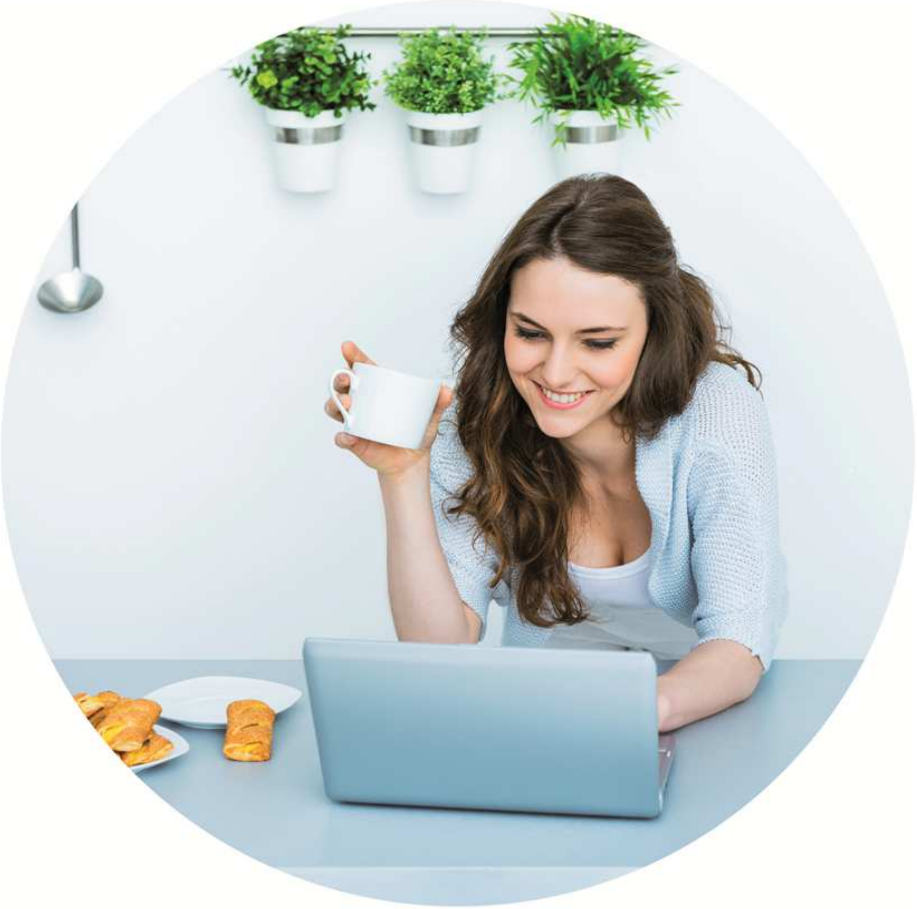
Sips Payment Acceptance Solution

Product Document

Sips Office JSON.WS



Public - DocId: WLSIPS.310.18.A.16R1.en

Contact: +33 (0) 811 10 70 33
sips@worldline.com / sips.worldline.com

Content

Version history 5
Associated documents 6
1 Introduction 7
1.1 About Worldline Sips 7
1.2 Purpose of this document 7
1.3 Prerequisites 8
2 Description of the protocol 9
2.1 REST protocol and JSON fields 9
2.1.1 Syntax of the request 9
2.1.2 Syntax for a list 10
2.1.3 Syntax of the Seal field 10
2.1.4 Syntax of concatenation for composite fields 11
2.1.5 Example of the Seal field computation 11
3 How to SIGN a MESSAGE 12
3.1 Reason for signing a message 12
3.2 Method used for signing a message 12
3.3 Code examples 12
3.3.1 Php 5 12
3.3.2 Java 13
3.3.3 .net 14
4 TESTING 15
4.1 Testing card transactions 15
5 How to START in PRODUCTION? 16
5.1 Merchant identifiers 16
5.2 Production validation 16
6 Service description 17
6.1 Description of the checkout service 17
6.1.1 cardOrder service 17
6.1.2 cardOrder example 27
6.1.3 walletOrder Service 27
6.1.4 WalletOrder example 29

6.1.5 directDebitOrder service 29
6.1.6 directDebitOrder example 32
6.1.7 paymentProviderInitialize service 33
6.1.8 paymentProviderInitialize example 35
6.1.9 POST form from the external wallet 36
6.1.10 paymentProviderFinalize service 36
6.1.11 paymentProviderFinalize example 38
6.1.12 creditTransferInquire service 39
6.1.13 creditTransferInquire example 40
6.1.14 creditTransferInitialize service 40
6.1.15 creditTransferInitialize example 42
6.1.16 POST form from the external bank transfer service 42
6.1.17 creditTransferFinalizeAndOrder service 43
6.1.18 creditTransferFinalizeAndOrder example 45
6.1.19 cardCheckEnrolLment service 45
6.1.20 cardCheckEnrollment example 47
6.1.21 walletCheckEnrollment service 47
6.1.22 walletCheckEnrollment example 49
6.1.23 POST form to the acs 50
6.1.24 cardValidateAuthenticationAndOrder service 51
6.1.25 cardValidateAuthenticationAndOrder example 53
6.1.26 cardValidateAuthentication service 54
6.1.27 cardValidateAuthentication example 55

6.2 Description of the cash management web service 56

6.2.1 Accept Challenge service 56
6.2.2 Example of an Accept Challenge operation 57
6.2.3 cancel service 57
6.2.4 Example of a cancellation operation 58
6.2.5 creditHolder service 58
6.2.6 Example of a credit holder operation 60
6.2.7 duplicate service 60
6.2.8 Example of a duplicate operation 63
6.2.9 referral service 64
6.2.10 Example of a referral operation 64
6.2.11 refund service 65
6.2.12 Example of a refund operation 66
6.2.13 Refuse Challenge service 66
6.2.14 Example of an Refuse Challenge operation 67
6.2.15 validation service 67
6.2.16 Example of a validation operation 68
6.2.17 recycle service 69
6.2.18 Example of a recycle operation 70
6.2.19 walletCreditHolder service 71
6.2.20 Example of awallet credit holder operation 72

6.3 Description of the diagnostic web service 73
6.3.1 *getTransactionData Service* 73
6.3.2 *Example of a getTransactionData information request* 75
6.4 Description of the wallet web service 76
6.4.1 *addCard service* 76
6.4.2 *Example of a addCard service* 77
6.4.3 *addDirectDebit service* 77
6.4.4 *Example of a addDirectDebit service* 78
6.4.5 *signOff service* 79
6.4.6 *Example of a signOff wallet deletion* 79
6.4.7 *updatePaymentMean service* 80
6.4.8 *Example of an update of a payment mean updatePaymentMean* 81
6.4.9 *deletePaymentMean service* 81
6.4.10 *Example of payment mean deletion deletePaymentMean* 82
6.4.11 *getWalletData service* 82
6.4.12 *Example of wallet reading getWalletData* 83
6.4.13 *Service getPaymentMeanData* 84
6.4.14 *Example of payment mean consultation getPaymentMeanData* 84
6.5 Description of the paymentMeanInfo web service 85
6.5.1 *getCardData service* 85
6.5.2 *Example of a card data consultation getCardData* 86
6.6 Description of the token web service 87
6.6.1 *tokenToPan service* 87
6.6.2 *Example of a tokenToPan information request* 88
6.6.3 *panToToken service* 88
6.6.4 *Example of a panToToken information request* 90
6.7 Description of the Fraud web service 90
6.7.1 *getVelocityData service* 90
6.7.2 *Example of a getVelocityData information request* 91

VERSION HISTORY

Date (aaaa/mm/dd)	Version	Description
2015/12/14	18	Update CardCheckEnrollment method to add panType field
2015/10/20	17	Update fraudData input Add new field sealAlgorithm in all services
2015/09/07	16	Add ShoppingCartDetail to CashManagement/refund service Add GetVelocityData operation of Fraud Service Add cardValidateAuthentication method in Checkout service
2015/07/17	15	Add forgotten panType field into CardOrder method Add new fields into PaymentProviderInitialize and CreditTransferInitialize operations Review .net implementation for seal calculation Add a description for the syntax of a list in JSON
2015/07/13	14	Add new fields into GetTransactionData and PaymentProviderInitialize operations
2015/05/05	13	Add TokenToPan and PanToToken operations Add WalletCreditHolder operation Add in Checkout Service 3D operations with CardCheckEnrollment, WalletCheckEnrollment and CardValidateAuthenticationAndOrder Add in Checkout credit transfer operations with CreditTransferInquire, CreditTransferInitialize and CreditTransferFinalizeAndOrder Add cardData fields to output
2015/04/08	12	Add RiskManagementCustomData and ShoppingCartDetail to checkout/cardOrder and checkout/DirectDebitOrder
2015/01/25	11	Add checkAVS field Fix an erorr in .dotnet example
2014/12/11	10	Add payment mean info service. Add accept and refused transaction
2014/09/05	09	Remove addSdd operation, replaced by addDirectDebit operation Change WS version from 2.0 to 2.5 in consistency with SOAP version Add PaymentMeanData to refund Add PaymentMeanData and holderData to cardOrder Fix interface version in cardOrder example Fix fromMerchantId from Mandatory to Optional in duplicate function
2014/05/20	08	Add Recycle operation, and some new fields in walletOrder, directDebitOrder, getTransactionData and duplicate operation Add paymentProviderInitialize, paymentProviderFinalize Add more explanation for complex fields and seal calculation
2014/05/20	07	Fix typo regarding http<>https Fix optional/mandatory fields in duplicate operation Fix getTransactionData output fields missing
2014/04/03	06	Add Wallet management
2014/03/14	05	Best explanation of keyVersion management Change .dotNet and Java examples
2014/03/05	04	Change validation to validate in production URL
2014/02/19	03	Change name to Sips office JSON
2014/02/06	02	Add missing URL for cashManagement
2013/12/05	01	Initial version

NB: only an "Approved" document will be considered valid.

ASSOCIATED DOCUMENTS

Doc iD	Reference
DiD01	WLSIPS.005 GD Glossary (in latest version, Approved status)

1 INTRODUCTION

1.1 ABOUT WORLDLINE SIPS

To better apprehend the solution, you are advised to visit http://sips.worldline.com/.

All terms, acronyms, expressions specific to Sips Payment Acceptance Solution context are defined in the document [DiD01].
Please refer to them as appropriate.

1.2 PURPOSE OF THIS DOCUMENT

The purpose of the present document is to explain how to implement exchanges between merchant websites and the Sips Office JSON solution.

This document is aimed at all merchants who wish to subscribe to the Sips offer and wish to use a connector based on JSON exchanges through REST protocol between merchant websites and the Sips servers, while using Office JSON as a gateway.

The **checkout** interface makes it possible to:

- create a transaction using a card, an electronic wallet or a SDD,ELV payment mean;
- initialize and finalize a transaction with an external wallet (Masterpass, Paylib or V.Me);
- manage 3D-Secure verification for card and electronic wallet transaction;

The **cash management** interface makes it possible to:

- cancel a transaction;
- duplicate a transaction;
- refund a transaction;
- force a transaction;
- validate a transaction.
- recycle a transaction
- accept or refused a transaction

The **diagnostic** interface makes it possible to:

- find information linked to a transaction

The **wallet** interface makes it possible to:

- create or delete a merchant wallet

- add, update or delete payment means from a merchant wallet

The **payment mean info** web service interface makes it possible to:

- consult the card information associated with a card number or a card IIN

The **token** web service interface makes it possible to:

- tokenize a pan or retrieve a tokenized pan

The **fraud** web service interface makes it possible to:

- check fraud on existing transactions

1.3 PREREQUISITES

Basic knowledge of the standards in web development languages in the industry such as Java, Php or .Net is necessary in order to develop a client that can connect to the Office JSON gateway.

This solution ensures that messages exchanged between the Merchant's website and the Sips servers are secured with the help of secret keys.

The Merchant is responsible for storing and managing these keys securely.

✍ **Note**

> If the key is compromised, or you suspect that it is, it is the responsibility of the Merchant to renew their secret key via the Sips Download extranet.

2 Description of the protocol

2.1 REST protocol and JSON fields

JSON is syntax for storing and exchanging text information. Much like XML, JSON is smaller than XML, and faster and easier to parse.

The simplest way to access any service is through cURL.

Example of a cURL request with PHP:

```
<php?

…

// Open cURL session and data are sent to server
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,            $url_of_service );
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true );
curl_setopt($ch, CURLOPT_POST,           true );
curl_setopt($ch, CURLOPT_POSTFIELDS,     $data_to_send);
curl_setopt($ch, CURLOPT_HTTPHEADER,     array('Content-Type: application/json', 'Accept:
application/json'));
curl_setopt($ch, CURLOPT_PROXY,          $name_and_proxy_port);
$result = curl_exec($ch);
$info = curl_getinfo($ch);

// Manage errors
if ($result == false || $info['http_code'] != 200) {
  echo $result;
  if (curl_error($ch))
    $result .= "\n". curl_error($ch);
}

// Close cURL session
curl_close($ch);
…
…

?>
```

In the following examples, the data that is passed with JSON request to server depends entirely on which service you are trying to access and what fields are required by that service.

For specific examples see the documentation provided by the individual services.

✍ **Note**

Field names are case sensitive.

2.1.1 Syntax of the request

The data field is built according to the following format:

{“<field name>” : ”<value name>”, “<field name>” : “<value name>”, “field name” : “value name” etc., “seal” : “value of seal” }

All the fields necessary for the transaction (please see details in the data dictionary) must be present in this character chain. The order of the fields is irrelevant.

Example of a card order request:

```
{"amount" : "1000", "captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"100","cardExpiryDate" : "201406","cardNumber" : "4975497549754975", "currencyCode" : "978",
"interfaceVersion" : "IR_WS_2.3","merchantId" : "011223744550001", "orderChannel" :
"INTERNET", "transactionReference" : "tref20131209",”keyVersion” : “1”, "seal" :
"112a4b079ece08a0a55511cd5469fc47051d6ddb1404623170ba3873668e5c58" }
```

2.1.2 SYNTAX FOR A LIST

The syntax for the making of a list in JSON syntax follows the standard. Please find a summary of this structure for the two main cases of lists, a list of simple fields (e.g. String) or a list of object:

- List of simple fields:

…,"field name" : ["value1","value2"],…

Example of a simple fields list:

```
{"amount" : "1000","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"100","cardExpiryDate" : "201406","cardNumber" : "4975497549754975","currencyCode" :
"978","fraudData" : {"bypassCtrlList" : ["All","ForeignBinCard"]},"interfaceVersion" :
"IR_WS_2.3","keyVersion" : "1","merchantId" : "011223744550001","orderChannel" :
"INTERNET","transactionReference" : "tref20131209","seal" :
"f18d95fcfbcc02e577467a9ddaa86b74d65908e9c36df12c74465f1324bf2100"}
```

- List of objects:

…,“field name” : [{“sub field name1”:”value1”,“sub field name2”:”value2”},{“sub field name1”:”value3”, “sub field name2”:”value4”}],…

Example:

```
{"amount" : "1000","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"100","cardExpiryDate" : "201604","cardNumber" : "4975497549754975","currencyCode" :
"978","interfaceVersion" : "IR_WS_2.3","keyVersion" : "1","merchantId" :
"011223744550001","orderChannel" : "INTERNET","riskManagementCustomDataList" :
[{"riskManagementCustomSequence" : "abc","riskManagementCustomValue" :
"123"},{"riskManagementCustomSequence" : "efg","riskManagementCustomValue" :
"456"}],"transactionReference" : "tref20131209","seal" :
"1e0f6cf0f2ca8365b84c9118f39ab3ed2837e114b9b42c1a49da1db368ce76d4"}
```

2.1.3 SYNTAX OF THE SEAL FIELD

The value of the Seal field is built as follows:

- Concatenation of the data fields **values** regarding **name of fields in alphabetical** order**, except keyVersion and sealAlgorithm fields**
- Obtaining the UTF-8 encoding of the data for the previous result

- HMAC with SHA256 encrypting of the bytes obtained with the secret key

This procedure can be summarised as follows:

2.1.4 HMAC(SHA256(UTF-8(SORTEDDATAVALUES,SECRETKEY)))SYNTAX OF CONCATENATION FOR COMPOSITE FIELDS

If you are going to use composite fields with substructure in the fields, you need to use a specific field calculation for that:

- Concatenation of composite field and the substructure fieldname with case sensitive respect
- Sort of all fields in substructure and then sort of these fields with the other fields

Example:

```
The field name to use for byPassCrtList of fraudData must be fraudDatabyPassCtrlList and
its value will be sorted with other fields.
If it contains All,ForeignBinCard, its value become AllForeignBinCard for computation
of seal.
```

2.1.5 EXAMPLE OF THE SEAL FIELD COMPUTATION

```
<?php

…

// Seal computation thanks to hash sorted data hash with merchant key
$data_to_send=$amount.$captureDay.$captureMode.$cardCSCValue.$cardExpiryDate.$cardNumber.$card
SeqNumber.$currency_code.$pb.$customerId.$customerIpAddress.$ivers.$mid.$merchantTransactionDa
teTime.$orderChannel.$orderId.$returnContext.$opeorig.$tref;


$data_to_send= utf8_encode($data_to_send)

$seal=hash_hmac('sha256', $data_to_send, $secretKey);

…
…

?>
```

3 How to Sign a Message

3.1 Reason for signing a message

The request contains the transaction settings and is sent through the Internet user's browser. It is theoretically possible for a hacker to intercept the request and change the settings before the data reaches the payment server.
Therefore, it is necessary to add security to ensure the integrity of the transaction settings sent. The Sips solution meets this need by exchanging signatures.
A successful signature check involves two things:

- The **integrity** of the request and response messages, no alteration during exchange
- The **authentication** of the issuer and receiver, as they share the same secret key.

✍ **Note**

If the key used for the signature is compromised, or you suspect that it is, it is the Merchant's responsibility to request the renewal of their key by connecting to Sips Download.

3.2 Method used for signing a message

The signature operation is completed by calculating the encrypted value based on the transaction's settings (the field values) to which the secret key (unknown by the Internet user) is added. All character chains are **converted to UTF8 before** being encrypted.

The encrypting algorithm (SHA256) produces an irreversible result. Generally, when such a message is received, the message receiver must recalculate the encrypted value in order to compare it with the value received. Any difference indicates that the data exchanged was falsified.

The result must be sent in hexadecimal form in the field named Seal.

3.3 Code examples

3.3.1 PHP 5

```
<?php
…
echo hash_mac('sha256', $data, $secretKey);
…
```

```
?>
```

data and secretKey must use a UTF-8 character set. Refer to the **utf8_encode** function to convert from ISO-8859-1 to UTF-8.

3.3.2 JAVA

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class ExampleHMACSHA256 {

        /**
         * table to convert a nibble to a hex char.
         */
        static final char[] hexChar = {
           '0' , '1' , '2' , '3' ,
           '4' , '5' , '6' , '7' ,
           '8' , '9' , 'a' , 'b' ,
           'c' , 'd' , 'e' , 'f'};

        /**
         * Fast convert a byte array to a hex string
         * with possible leading zero.
         * @param b array of bytes to convert to string
         * @return hex representation, two chars per byte.
         */
        public static String encodeHexString ( byte[] b )
           {
           StringBuffer sb = new StringBuffer( b.length * 2 );
           for ( int i=0; i<b.length; i++ )
              {
              // look up high nibble char
              sb.append( hexChar [( b[i] & 0xf0 ) >>> 4] );

              // look up low nibble char
              sb.append( hexChar [b[i] & 0x0f] );
              }
           return sb.toString();
           }

        /**
         * Computes the seal
         * @param Data the parameters to cipher
         * @param secretKey the secret key to append to the parameters
         * @return hex representation of the seal, two chars per byte.
         */
        public static String computeSeal(String Data, String secretKey) throws Exception
        {
          Mac hmacSHA256;

          hmacSHA256 = Mac.getInstance("HmacSHA256");
          SecretKeySpec keySpec = new SecretKeySpec(secretKey, "HmacSHA256");
          hmacSHA256.init(keySpec);

          return encodeHexString(hmacSHA256.doFinal(Data));
        }

        /**
         * @param args
         */
        public static void main(String[] args) {
                try {
                        System.out.println (computeSeal("parameters", "key"));
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

}
```

3.3.3 .NET

(Completed using a simple form called "Form 1" containing two text fields for entering: `txtSips`, `txtSecretKey` and another for displaying: `lblHEX`)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Security.Cryptography;

namespace ExampleDotNET
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdGO_Click(object sender, EventArgs e)
        {
            String sChaine = txtSips.Text;
            UTF8Encoding utf8 = new UTF8Encoding();
            Byte[] encodedBytes = utf8.GetBytes(sChaine);

            byte[] shaResult;

            HMAC hmac = new HMAC.Create("HMACSHA256");
            var key = "YourSecretKey";
            hmac.Key = utf8.GetBytes(key);
            hmac.Initialize();

            shaResult = hmac.ComputeHash(encodedBytes);

            lblHEX.Text = ByteArrayToHEX(shaResult);
        }

        private string ByteArrayToHEX(byte[] ba)
        {
            StringBuilder hex = new StringBuilder(ba.Length * 2);
            foreach (byte b in ba)
                hex.AppendFormat("{0:x2}", b);
            return hex.ToString();
        }

    }
}
```

4 TESTING

The testing and integration stages can be completed using the user acceptance test environment.

In order to be able to use this environment, you must contact technical support, indicating the services required. Support will send you a merchant identifier (merchantID) and a secret key.

The technical details required to use this environment are described below:

URL of the user acceptance	https://office-server.test.sips-atos.com
Name of the payment service	/rs-services/v2/checkout
Name of the cash management service	/rs-services/v2/cashManagement
Name of the diagnostic service	/rs-services/v2/diagnostic
Name of the wallet service	/rs-services/v2/wallet
Name of the payment mean info service	/rs-services/v2/paymentMeanInfo
Name of the token service	/rs-services/v2/token
Name of the fraud service	/rs-services/v2/fraud

In the user acceptance environment, the authorisation process is simulated. This means that it is not necessary to use the real payment methods in order to complete these tests.

4.1 TESTING CARD TRANSACTIONS

You must contact technical support to obtain the usable card numbers for user acceptance test environment.

This list will allow to do all test cases for every Sips response code and acquirer response codes.

For iDEAL and SDD payment means, please contact technical support.

5 How to START in PRODUCTION?

The next step is to connect to the production environment for the actual go-live.

In order to do this, the Merchant must change the office server URL and use the Merchant IDs received during the registration stage.

5.1 Merchant IDENTIFIERS

The URL for the production payment server is: https://office-server.sips-atos.com/

To access the production environment, you will need the following two pieces of information:

- The merchant identifier (**merchantID)** which identifies the eCommerce site on the Sips server
- The version (**keyVersion)** of the secret key
- The secret key (**secretKey)** used to sign requests and verify responses

The merchant identifier (**merchantID)** is provided by technical support at the end of the merchant's registration stage.

You can download the version of the key (**keyVersion)** and the secret key (**secretKey)** from the extranet https://download.sips-atos.com using the username and password provided by technical support at the end of the Merchant registration stage.

5.2 Production VALIDATION

From the point when the merchant uses its own identifiers on the production server, any cash management transactions and operations performed are real (for example, payment transactions are real from end-to-end, up until the funds are credited to the merchant's account and debited from the buyer's account).

Before the shop is actually opened to the public, the merchant may submit a request to validate the end-to-end payment, up until the funds are credited to the merchant's account and debited from the buyer's account

6 SERVICE DESCRIPTION

6.1 DESCRIPTION OF THE CHECKOUT SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/checkout`

This service allows you to complete a payment order using a card or electronic wallet.

6.1.1 CARDORDER SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/checkout/cardOrder`

Requests for payment orders via card include the following elements:

- Default inputs
- authenticationData inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- holderAddress inputs
- holderContact inputs
- holderData inputs
- paymentMeanData inputs
- riskManagementCustomData inputs
- s10TransactionReference inputs
- shoppingCartDetail inputs

6.1.1.1 Default inputs for the cardOrder

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
captureDay	Optional	WS_2.3	
captureMode	Optional	WS_2.3	
cardCSCValue	Optional	WS_2.3	
cardEffectiveDate	Optional	WS_2.3	
cardExpiryDate	Mandatory	WS_2.3	
cardNumber	Mandatory	WS_2.3	
cardSeqNumber	Optional	WS_2.3	

Field	Presence	As of version	Comments
currencyCode	Mandatory	WS_2.3	
customerId	Optional	WS_2.3	
customerIpAddress	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at IR_WS_2.9
invoiceReference	Optional	WS_2.3	
merchantId	Mandatory	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantTransactionDateTime	Optional	WS_2.3	
orderChannel	Optional	WS_2.3	Optional for AccordKdo Card, mandatory for all other payment means
orderId	Optional	WS_2.3	
panType	Optional	WS_2.4	
paymentMeanData	Optional	WS_2.5	
paymentPattern	Optional	WS_2.3	This field is mandatory for certain payment means. Please refer to the respective payment mean implementation guide for details.
returnContext	Optional	WS_2.3	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.3	
transactionReference	Optional	WS_2.6	
transactionOrigin	Optional	WS_2.3	
billingAddress	Optional	WS_2.3	See below
billingContact	Optional	WS_2.3	See below
customerAddress	Optional	WS_2.3	See below
customerContact	Optional	WS_2.3	See below
deliveryAddress	Optional	WS_2.3	See below
deliveryContact	Optional	WS_2.3	See below
authenticationData	Optional	WS_2.3	See below
customerData	Optional	WS_2.3	See below
fraudData	Optional	WS_2.3	See below
holderAddress	Optional	WS_2.3	See below
holderContact	Optional	WS_2.3	See below
holderData	Optional	WS_2.5	See below
paymentMeanData	Optional	WS_2.5	See below
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
riskManagementCustomDataList	Optional	WS_2.8	See below
shoppingCartDetail	Optional	WS_2.8	See below

Table 1: Fields for cardOrder request

6.1.1.2 s10TransactionReference inputs

Field	Presence	As of version	Comments
s10TransactionId	Optional	WS_2.6	
s10TransactionIdDate	Optional	WS_2.6	

Table 2: Fields for the s10TransactionReference element

6.1.1.3 billingAddress inputs

Field	Presence	As of version	Comments
addressAdditional1	Optional	WS_2.3	
addressAdditional2	Optional	WS_2.3	
addressAdditional3	Optional	WS_2.3	
city	Optional	WS_2.3	
company	Optional	WS_2.3	
country	Optional	WS_2.3	
postBox	Optional	WS_2.3	
state	Optional	WS_2.3	
street	Optional	WS_2.3	
streetNumber	Optional	WS_2.3	
zipCode	Optional	WS_2.3	

Table 3: Fields for the billingAddress element

6.1.1.4 billingContact inputs

Field	Presence	As of version	Comments
email	Optional	WS_2.3	
firstname	Optional	WS_2.3	
gender	Optional	WS_2.3	
lastname	Optional	WS_2.3	
mobile	Optional	WS_2.3	
phone	Optional	WS_2.3	
title	Optional	WS_2.3	

Table 4: Fields for the billingContact element

6.1.1.5 customerAddress inputs

Field	Presence	As of version	Comments
addressAdditional1	Optional	WS_2.3	
addressAdditional2	Optional	WS_2.3	
addressAdditional3	Optional	WS_2.3	
city	Optional	WS_2.3	
company	Optional	WS_2.3	
country	Optional	WS_2.3	
postBox	Optional	WS_2.3	
state	Optional	WS_2.3	
street	Optional	WS_2.3	
streetNumber	Optional	WS_2.3	
zipCode	Optional	WS_2.3	

Table 5: Fields for the customerAddress element

6.1.1.6 customerContact inputs

Field	Presence	As of version	Comments
email	Optional	WS_2.3	
firstname	Optional	WS_2.3	
gender	Optional	WS_2.3	
lastname	Optional	WS_2.3	
mobile	Optional	WS_2.3	
phone	Optional	WS_2.3	
title	Optional	WS_2.3	

Table 6: Fields for the customerContact element

6.1.1.7 deliveryAddress inputs

Field	Presence	As of version	Comments
addressAdditional1	Optional	WS_2.3	
addressAdditional2	Optional	WS_2.3	
addressAdditional3	Optional	WS_2.3	
City	Optional	WS_2.3	
company	Optional	WS_2.3	
country	Optional	WS_2.3	
postBox	Optional	WS_2.3	
state	Optional	WS_2.3	
street	Optional	WS_2.3	

Field	Presence	As of version	Comments
streetNumber	Optional	WS_2.3	
zipCode	Optional	WS_2.3	

Table 7: Fields for the deliveryAddress element

6.1.1.8 deliveryContact inputs

Field	Presence	As of version	Comments
email	Optional	WS_2.3	
firstname	Optional	WS_2.3	
gender	Optional	WS_2.3	
lastname	Optional	WS_2.3	
mobile	Optional	WS_2.3	
phone	Optional	WS_2.3	
Title	Optional	WS_2.3	

Table 8: Fields for the deliveryContact element

6.1.1.9 fraudData inputs

Field	Presence	As of version	Comments
allowedCardArea	Optional	WS_2.3	Future use
allowedCardCountryList	Optional	WS_2.3	
allowedIpArea	Optional	WS_2.3	Future use
allowedIpCountryList	Optional	WS_2.3	
bypass3DS	Optional	WS_2.3	Future use
bypassCtrlList	Optional	WS_2.3	
bypassInfoList	Optional	WS_2.3	
deniedCardArea	Optional	WS_2.3	Future use
deniedCardCountryList	Optional	WS_2.3	
deniedIpArea	Optional	WS_2.3	Future use
deniedIpCountryList	Optional	WS_2.3	
riskManagementDynamicSettingList	Optional	WS_2.10	

Table 9: Fields for the fraudData element

6.1.1.10 riskManagementDynamicSetting inputs

Field	Presence	As of version	Comments
riskManagementDynamicParam riskManagementDynamicValue	Mandatory Mandatory	WS_2.10 WS_2.10	

Table 10: Fields for the riskManagementDynamicSetting element

6.1.1.11 holderAddress inputs

Field	Presence	As of version	Comments
addressAdditional1	Optional	WS_2.3	
addressAdditional2	Optional	WS_2.3	
addressAdditional3	Optional	WS_2.3	
city	Optional	WS_2.3	
company	Optional	WS_2.3	
country	Optional	WS_2.3	
postBox	Optional	WS_2.3	
state	Optional	WS_2.3	
street	Optional	WS_2.3	
streetNumber	Optional	WS_2.3	
zipCode	Optional	WS_2.3	

Table 11: Fields for the holderAddress element

6.1.1.12 holderContact inputs

Field	Presence	As of version	Comments
email	Optional	WS_2.3	
firstname	Optional	WS_2.3	
gender	Optional	WS_2.3	
lastname	Optional	WS_2.3	
mobile	Optional	WS_2.3	
phone	Optional	WS_2.3	
title	Optional	WS_2.3	

Table 12: Fields for the holderContact element

6.1.1.13 holderData inputs

Field	Presence	As of version	Comments
birthCity	Optional	WS_2.5	Future use
birthCountry	Optional	WS_2.5	Future use
birthDate	Optional	WS_2.5	
birthZipCode	Optional	WS_2.5	Future use
nationalityCountry	Optional	WS_2.5	Future use

newPwd	Optional	WS_2.5	Future use
pwd	Optional	WS_2.5	Future use

Table 13: Fields for the holderContact element

6.1.1.14 customerData inputs

Field	Presence	As of version	Comments
birthCity	Optional	WS_2.3	Future use
birthCountry	Optional	WS_2.3	Future use
birthDate	Optional	WS_2.3	
birthZipCode	Optional	WS_2.3	Future use
nationalityCountry	Optional	WS_2.3	Future use
newPwd	Optional	WS_2.3	Future use
pwd	Optional	WS_2.3	Future use

Table 14: Fields for the customerData element

6.1.1.15 authenticationData inputs

The authenticationData input includes the following elements:

- cardAuthPolicy inputs
- threeD inputs

Field	Presence	As of version	Comments
ignoreCSCCheckResult	Optional	WS_2.3	
ignorePostcodeCheckResult	Optional	WS_2.3	
ignoreAddressCheckResult	Optional	WS_2.3	
automaticReverse	Optional	WS_2.3	
checkAVS	Optional	WS_2.7	

Table 15: Fields for the authenticationData/cardAuthPolicy element

Field	Presence	As of version	Comments
cavv	Optional	WS_2.3	
cavvAlgorithm	Optional	WS_2.3	
Eci	Optional	WS_2.3	
securityIndicator	Optional	WS_2.3	
txStatus	Optional	WS_2.3	
xid	Optional	WS_2.3	

Table 16: Fields for the authenticationData/threeD element

Only the securityIndicator field is mandatory in the case of a 3D payment.

6.1.1.16 paymentMeanData inputs

Field	Presence	As of version	Comments
accord	Optional	WS_2.5	See below
accordkdo	Optional	WS_2.5	See below

Table 17: Fields for the paymentMeanData element

6.1.1.17 accord inputs

Field	Presence	As of version	Comments
settlementMode	Optional	WS_2.5	
additionalAuthorisationNumber	Optional	WS_2.5	

Table 18: Fields for the accord element

6.1.1.18 accordkdo inputs

Field	Presence	As of version	Comments
settlementMode	Optional	WS_2.5	

Table 19: Fields for the accordkdo element

6.1.1.19 shoppingCartDetail inputs

The shoppingCartDetail input includes the following elements:

- shoppingCartItem inputs

Field	Presence	As of version	Comments
shoppingCartDetail.shoppingCartTotalAmount	Optional	WS_2.8	
shoppingCartDetail.shoppingCartTotalQuantity	Optional	WS_2.8	
shoppingCartDetail.shoppingCartTotalTaxAmount	Optional	WS_2.8	
shoppingCartDetail.mainProduct	Optional	WS_2.8	
shoppingCartDetail.shoppingCartItemList	Optional	WS_2.8	

Table 20: Fields for the shoppingCartDetail element

6.1.1.20 shoppingCartItem inputs

Field	Presence	As of version	Comments
shoppingCartItem.productName	Optional	WS_2.8	
shoppingCartItem.productDescription	Optional	WS_2.8	
shoppingCartItem.productCode	Optional	WS_2.8	
shoppingCartItem.productSKU	Optional	WS_2.8	
shoppingCartItem.productUnitAmount	Optional	WS_2.8	
shoppingCartItem.productQuantity	Optional	WS_2.8	
shoppingCartItem.productTaxRate	Optional	WS_2.8	
shoppingCartItem.productUnitTaxAmount	Optional	WS_2.8	
shoppingCartItem.productCategory	Optional	WS_2.8	

Table 21: Fields for the shoppingCartItem element

6.1.1.21 riskManagementCustomData inputs

Field	Presence	As of version	Comments
riskManagementCustomData.riskManagementCustomSequence	Optional	WS_2.8	
riskManagementCustomData.riskManagementCustomValue	Optional	WS_2.8	

Table 22: Fields for the riskManagementCustomData element

6.1.1.22 cardOrder outputs

Response for payment orders via card includes the following elements:

- Default outputs
- cardData outputs
- paymentMeanData outputs
- s10TransactionReference outputs

Field	As of version	Comments
acquirerResponseCode	WS_2.3	Only if refused during the pre-authorisation check
authorisationId	WS_2.3	Only if approved.
avsAddressResponseCode*	WS_2.3	
avsPostcodeResponseCode*	WS_2.3	
cardCSCResultCode	WS_2.3	

Field	As of version	Comments
cardData	WS_2.9	See 6.1.1.23
cardScheme	WS_2.3	
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
guaranteeIndicator	WS_2.3	
holderAuthentRelegationCode*	WS_2.3	
holderAuthentStatus*	WS_2.3	
maskedPan	WS_2.3	
recurringResponseCode*	WS_2.3	
responseCode	WS_2.3	
returnContext	WS_2.3	Value transferred to the payment request.
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreColor*	WS_2.3	
scoreInfo*	WS_2.3	
scoreProfile*	WS_2.3	
scoreThreshold*	WS_2.3	
scoreValue*	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	
tokenPan	WS_2.3	
paymentMeanData	WS_2.5	See 6.1.1.16

Table 23: Fields for cardOrder response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.1.23 cardData outputs

Field	Presence	As of version	Comments
cardProductCode	Optional	WS_2.9	
cardProductName	Optional	WS_2.9	
cardProductProfile	Optional	WS_2.9	
cardScheme	Optional	WS_2.9	
issuerCode	Optional	WS_2.9	
issuerCountryCode	Optional	WS_2.9	

Table 24: Fields for the cardData element

6.1.2 CARDORDER EXAMPLE

6.1.2.1 Example of cardOrder request

```
{"amount" : "2500","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"0000","cardExpiryDate" : "201612","cardNumber" : "1234123412341234","currencyCode" :
"978","interfaceVersion" : " IR_WS_2.3","keyVersion" : "1","merchantId" :
"011223344550000","orderChannel" : "INTERNET","orderId" : " ORD101","returnContext" : "
ReturnContext","transactionOrigin" : " SO_WEBAPPLI","transactionReference" :
"TREFEXA2012",","paymentMeanData" : {"sdd" : {"mandateAuthentMethod" :
"SMS_OTP","mandateUsage" : "ONE_OFF","mandateId" : "000000000000000283"}},"seal" :
"2205f0636dc500c4f3ef536075895b8baba3a60c7087e06cd9d330c50a50c53e"}
```

6.1.2.2 Example of cardOrder response

```
{"acquirerResponseCode":"00","authorisationId":"123456789102","cardScheme":"MASTERCARD","compl
ementaryCode":"","complementaryInfo":"","maskedPan":"5219##########00","responseCode":"00","re
turnContext":"ReturnContext","transactionDateTime":"2012-10-16T23:50:44-
12:00","seal":"7ca065ebba90762e3286ee16d6a2d2da0eea8bae6b46592058933881d96d0c3d"}
```

6.1.3 WALLETORDER SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/checkout/walletOrder`

Requests for payment orders via electronic wallets include the following elements:

- Default inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- holderAddress inputs
- holderContact inputs
- customerData inputs
- authenticationData inputs
- s10TransactionReference inputs

6.1.3.1 Default inputs for the WalletOrder

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
captureDay	Optional	WS_2.3	
captureMode	Optional	WS_2.3	
currencyCode	Mandatory	WS_2.3	
customerId	Optional	WS_2.3	
customerIpAddress	Optional	WS_2.3	
cardCSCValue	Optional	WS_2.3	

Field	Presence	As of version	Comments
invoiceReference	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at IR_WS_2.9
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
merchantTransactionDateTime	Optional	WS_2.3	
merchantWalletId	Mandatory	WS_2.3	
orderChannel	Mandatory	WS_2.3	
orderId	Optional	WS_2.3	
paymentMeanId	Mandatory	WS_2.3	
returnContext	Optional	WS_2.3	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.3	
transactionReference	Optional	WS_2.6	
transactionOrigin	Optional	WS_2.3	
paymentPattern	Optional	WS_2.3	This field is mandatory for certain payment means. Please refer to the respective payment mean implementation guide for details.
billingAddress	Optional	WS_2.3	Same as cardOrder
billingContact	Optional	WS_2.3	Same as cardOrder
customerAddress	Optional	WS_2.3	Same as cardOrder
customerContact	Optional	WS_2.3	Same as cardOrder
customerData	Optional	WS_2.3	Same as cardOrder
deliveryAddress	Optional	WS_2.3	Same as cardOrder
deliveryContact	Optional	WS_2.3	Same as cardOrder
holderAddress	Optional	WS_2.3	Same as cardOrder
holderContact	Optional	WS_2.3	Same as cardOrder
customerData	Optional	WS_2.3	Same as cardOrder
authenticationData	Optional	WS_2.3	Same as cardOrder
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2

Table 25: Fields for WalletOrder request

6.1.3.2 Wallet order output

Responses for payment orders via electronic wallets include the following elements:

- Default outputs
- cardData outputs
- s10TransactionReference outputs

Field	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
cardData	WS_2.9	See 6.1.1.23
cardScheme	WS_2.3	
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
maskedPan	WS_2.3	
responseCode	WS_2.3	
returnContext	WS_2.3	Value transferred to the payment request.
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreColor*	WS_2.1	
scoreInfo*	WS_2.1	
scoreProfile*	WS_2.1	
scoreThreshold*	WS_2.1	
scoreValue*	WS_2.1	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	
tokenPan	WS_2.1	

Table 26: Fields for WalletOrder response

6.1.4 WALLETORDER EXAMPLE

6.1.4.1 Example of walletOrder request

```
{"amount" : "2500","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","currencyCode" :
"978","customerIpAddress" : "127.0.0.1","interfaceVersion" : "IR_WS_2.3","invoiceReference" :
"FAC007NB","keyVersion" : "1","merchantId" : "011223344550000","merchantTransactionDateTime"
: "2014-01-10T17:13:30.743+01:00","orderChannel" : "INTERNET","merchantWalletId" :
"WALLET01", "paymentMeanId","10","transactionOrigin" : "SO_WEBAPPLI","transactionReference" :
"TREFEXA2012", "seal" : "5bb0e9fcbb3eaa28557ac41ed6ab57ab84d9edbe9fe8eb0cedd3e8548433b7cd"}
```

6.1.4.2 Example of walletOrder response

```
{"acquirerResponseCode":"00","authorisationId":"123456789102","cardScheme":"MASTERCARD","maske
dPan":"5219##########00","responseCode":"00","returnContext":"ReturnContext","transactionDateT
ime":"2012-10-16T23:50:44-
12:00","seal":"7ca065ebba90762e3286ee16d6a2d2da0eea8bae6b46592058933881d96d0c3d"}
```

6.1.5 DIRECTDEBITORDER SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/checkout/directDebitOrder`

Requests for payment orders via SDD or ELV include the following elements:

- Default inputs

- authenticationData inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- holderAddress inputs
- holderContact inputs
- instalmentData inputs
- paymentMeanData inputs
- riskManagementCustomData inputs
- s10TransactionReference inputs

6.1.5.1 Default inputs for the directDebitOrder

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
customerAccount	Optional	WS_2.5	
captureDay	Optional	WS_2.3	
captureMode	Optional	WS_2.3	
currencyCode	Mandatory	WS_2.3	
customerBankAccountType	Optional	WS_2.5	
customerBankCode	Optional	WS_2.5	
customerId	Optional	WS_2.3	
customerIpAddress	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at IR_WS_2.3
invoiceReference	Optional	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
merchantTransactionDateTime	Optional	WS_2.3	
orderChannel	Mandatory	WS_2.3	
orderId	Optional	WS_2.3	
paymentPattern	Optional	WS_2.3	This field is mandatory for certain payment means. Please refer to the respective payment mean implementation guide for details.
returnContext	Optional	WS_2.3	
paymentMeanBrand	Mandatory	WS_2.5	ELV or SEPA_DIRECT_DEBIT
sealAlgorithm	Optional	WS_2.11	

Field	Presence	As of version	Comments
statementReference	Optional	WS_2.3	
transactionReference	Optional	WS_2.6	
transactionOrigin	Optional	WS_2.3	
valueDate	Optional	WS_2.3	YYYYMMDD
billingAddress	Optional	WS_2.3	See cardOrder
billingContact	Optional	WS_2.3	See cardOrder
customerAddress	Optional	WS_2.3	See cardOrder
customerContact	Optional	WS_2.3	See cardOrder
deliveryAddress	Optional	WS_2.3	See cardOrder
deliveryContact	Optional	WS_2.3	See cardOrder
authenticationData	Optional	WS_2.3	See cardOrder
customerData	Optional	WS_2.3	See cardOrder
fraudData	Optional	WS_2.3	See cardOrder
holderAddress	Optional	WS_2.3	See cardOrder
holderContact	Optional	WS_2.3	See cardOrder
instalmentData	Optional	WS_2.3	See below
mandateAuthentMethod	M	WS_2.3	
mandateUsage	M	WS_2.3	
mandateId	Optional	WS_2.3	Mandatory for SEPA_DIRECT_DEBIT and optional for ELV
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
riskManagementCustomDataList	Optional	WS_2.8	See cardOrder
shoppingCartDetail	Optional	WS_2.8	See cardOrder

Table 27: Fields for directDebitOrder request

6.1.5.2 instalmentData inputs

Field	Presence	As of version	Comments
number	Optional	WS_2.3	
datesList	Optional	WS_2.3	
transactionReferencesList	Optional	WS_2.3	
s10TransactionIdsList	Optional	WS_2.6	
amountsList	Optional	WS_2.3	

Table 28: Fields for the instalmentData element

6.1.5.3 directDebitOrder outputs

Responses for payment orders via via SDD or ELV include the following elements:

- Default outputs
- S10TransactionReference outputs

Field	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
captureLimitDate	WS_2.3	
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
creditorId	WS_2.5	
errorFieldName	WS_2.3	
guaranteeIndicator *	WS_2.3	
holderAuthentRelegationCode*	WS_2.3	
holderAuthentStatus*	WS_2.3	
mandateId	WS_2.3	
maskedPan	WS_2.5	
responseCode	WS_2.3	
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreColor*	WS_2.3	
scoreInfo*	WS_2.3	
scoreProfile*	WS_2.3	
scoreThreshold*	WS_2.3	
scoreValue*	WS_2.3	
transactionActors	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	
valueDate	WS_2.5	

Table 29: Fields for directDebitOrder response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.6 DIRECTDEBITORDER EXAMPLE

6.1.6.1 Example of directDebitOrder request

```
{"amount" : "2500","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"0000","cardExpiryDate" : "201612","cardNumber" : "1234123412341234","currencyCode" :
"978","interfaceVersion" : " IR_WS_2.3","keyVersion" : "1","merchantId" :
"011223344550000","orderChannel" : "INTERNET","orderId" : " ORD101", "paymentMeanBrand" :
"SEPA_DIRECT_DEBIT" ,"returnContext" : "ReturnContext","transactionOrigin" : "
SO_WEBAPPLI","transactionReference" : "TREFEXA2012","seal" :
"2205f0636dc500c4f3ef536075895b8baba3a60c7087e06cd9d330c50a50c53e"}
```

6.1.6.2 Example of directDebitOrder response

```
{"acquirerResponseCode":"00","authorisationId":"123456789102","maskedPan":"5219##########00",
"responseCode":"00","transactionDateTime" : "2012-10-16T23:50:44-12:00","captureLimitDate" :
"2012-11-16T23:50:44-12:00","transactionActors" :
"BTOB","seal":"7ca065ebba90762e3286ee16d6a2d2da0eea8bae6b46592058933881d96d0c3d"}
```

6.1.7 PAYMENTPROVIDERINITIALIZE SERVICE

The URL is : `https://office-server.sips-atos.com/rs-services/v2/checkout/paymentProviderInitialize`

Requests for payment initialization for external wallet (Masterpass, Paylib or V.Me).

This request initializes a session for an external wallet order. For a non-mobile transaction, if the initialization step is successful, the merchant will receive in response a secure URL (redirectionUrl) to which the customer should be redirected to continue the payment order. This redirection must be made via a POST form.

In the request, the merchant shall also indicate the URL (merchantReturnUrl) to which the customer will be redirected to at the end of the external wallet payment. The merchant must then call the paymentProviderFinalize service to finalize the transaction.

Requests include the following elements:

- Default inputs
- authenticationData inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- paymentMeanData inputs
- s10TransactionReference inputs
- deliveryData inputs
- shoppingCartDetail inputs

6.1.7.1 Default inputs for the paymentProviderInitialize

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
captureDay	Optional	WS_2.3	
captureMode	Optional	WS_2.3	
currencyCode	Mandatory	WS_2.3	
customerId	Optional	WS_2.3	
customerIpAddress	Optional	WS_2.3	

Field	Presence	As of version	Comments
customerLanguage	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at IR_WS_2.3
invoiceReference	Optional	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
merchantReturnUrl	Mandatory	WS_2.3	
merchantTransactionDateTime	Optional	WS_2.3	
orderChannel	Mandatory	WS_2.3	
orderId	Optional	WS_2.3	
paymentMeanBrand	Mandatory	WS_2.3	
paymentPattern	Optional	WS_2.3	This field is mandatory for certain payment means. Please refer to the respective payment mean implementation guide for details.
returnContext	Optional	WS_2.3	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.3	
transactionReference	Optional	WS_2.6	
transactionOrigin	Optional	WS_2.3	
billingAddress	Optional	WS_2.3	See cardOrder
billingContact	Optional	WS_2.3	See cardOrder
customerAddress	Optional	WS_2.3	See cardOrder
customerContact	Optional	WS_2.3	See cardOrder
deliveryAddress	Optional	WS_2.3	See cardOrder
deliveryContact	Optional	WS_2.3	See cardOrder
authenticationData	Optional	WS_2.3	See cardOrder
customerData	Optional	WS_2.3	See cardOrder
paymentMeanData	Optional	WS_2.10	See below
fraudData	Optional	WS_2.3	See cardOrder
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
deliveryData	Optional	WS_2.6	See 6.2.7.2
shoppingCartDetail	Optional	WS_2.6	See 6.2.7.4
responseKeyVersion	Optional	WS_2.7	

Table 30: Fields for paymentProviderInitialize request

6.1.7.2 paymentMeanData inputs

Field	Presence	As of version	Comments
paypal	Optional	WS_2.10	See below

Table 31: Fields for the paymentMeanData element

6.1.7.3 paypal inputs

Field	Presence	As of version	Comments
landingPage	Optional	WS_2.10	
addrOverride	Optional	WS_2.10	
invoiceId	Optional	WS_2.10	
dupFlag	Optional	WS_2.10	
dupDesc	Optional	WS_2.10	
dupCustom	Optional	WS_2.10	
dupType	Optional	WS_2.10	

Table 32: Fields for the accord element

6.1.7.4 paymentProviderInitialize outputs

Responses include the following elements:

- Default outputs

Field	As of version	Comments
responseCode	WS_2.3	
redirectionUrl	WS_2.3	
redirectionData	WS_2.3	
messageVersion	WS_2.3	
paymentProviderSessionId	WS_2.3	
outerRedirectionUrl	WS_2.9	
acquirerNativeResponseCode	WS_2.10	
acquirerResponseIdentifier	WS_2.10	
acquirerResponseMessage	WS_2.10	

Table 33: Fields for paymentProviderInitialize response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.8 PAYMENTPROVIDERINITIALIZE EXAMPLE

6.1.8.1 Example of paymentProviderInitialize request

```
{"amount" : "1000","captureMode" : "AUTHOR_CAPTURE","currencyCode" :
"978","customerIpAddress" : "127.0.0.1","interfaceVersion" : "IR_WS_2.3","keyVersion" :
```

```
"1","merchantId" : "023101122334455","merchantReturnUrl" :
"http://www.yoursiteweb.com","merchantTransactionDateTime" : "2014-07-
11T10:59:57.723+02:00","orderChannel" : "INTERNET","paymentMeanBrand" : "PAYLIB","
transactionReference" : "TREFEXA2012","seal" : "
95e479c276d71a07c655f3c2db1bb8b483c0921e544d4af4cff3e220b04eb551"}
```

6.1.8.2 Example of paymentProviderInitialize response

```
{"responseCode":"00","redirectionUrl":"https://payment.sips.atos.fr/paymentprovider/init","red
irectionData":"RSs7cZyL1DqsVso","messageVersion":"0.1","seal":"79cf30535e08b8c6c9cfdd55c343845
947f8fea892ec0ea1db96378055f71de6"}
```

6.1.9 POST FORM FROM THE EXTERNAL WALLET

Once the payment order is processed, the customer is redirected to the merchant website. This redirection sends a POST form to the URL indicated in merchantReturnUrl at the paymentProviderInitialize request.
The merchant must retrieve the fields and send them to the paymentProviderFinalized request.

6.1.9.1 POST form outputs

Field	As of version	Comments
merchantId	WS_2.3	
messageVersion	WS_2.3	
redirectionData	WS_2.3	
transactionReference	WS_2_2	

Table 34: Fields for POST form response

6.1.10 PAYMENTPROVIDERFINALIZE SERVICE

Requests for payment finalization for external wallet (Masterpass, Paylib or V.Me):

This request is mandatory to know the result of the payment order with an external wallet. It must be called after the merchant received the POST form from the external wallet through the merchantReturnUrl (see paymentProviderInitialize request).

Requests include the following elements:

- Default inputs
- s10TransactionReference inputs

6.1.10.1 Default inputs for the paymentProviderFinalize

Field	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.3	Value fixed at IR_WS_2.6
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	

Field	Presence	As of version	Comments
messageVersion	Mandatory	WS_2.3	
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	
redirectionData	Mandatory	WS_2.3	

Table 35: Fields for paymentProviderFinalize request

6.1.10.2 paymentProviderFinalize outputs

Responses include the following elements:

- Default outputs
- cardData outputs
- s10TransactionReference outputs

	As of version	Comments
acquirerNativeResponseCode	WS_2.10	
acquirerResponseCode	WS_2.3	
acquirerResponseIdentifier	WS_2.10	
acquirerResponseMessage	WS_2.6	
avsAddressResponseCode	WS_2.3	
avsPostcodeResponseCode	WS_2.3	
authorisationId	WS_2.3	
captureLimitDate	WS_2.3	
cardData	WS_2.9	See 6.1.1.23
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
currencyCode	WS_2.3	
captureDay	WS_2.3	
captureMode	WS_2.3	
cardExpiryDate	WS_2.3	
cardCSCResultCode	WS_2.3	
customerId	WS_2.3	
customerIpAddress	WS_2.3	
guaranteeIndicator *	WS_2.3	
holderAuthentMethod*	WS_2.3	
holderAuthentStatus*	WS_2.3	
holderAuthentProgram*	WS_2.3	
invoiceReference	WS_2.3	
issuerWalletInformation	WS_2.6	

	As of version	Comments
maskedPan	WS_2.3	
merchantId	WS_2.3	
merchantTransactionDateTime	WS_2.3	
orderChannel	WS_2.3	
orderId	WS_2.6	
panEntryMode	WS_2.3	
paymentMeanBrand	WS_2.3	
paymentPattern	WS_2.3	
responseCode	WS_2.3	
returnContext	WS_2.3	
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreColor*	WS_2.3	
scoreInfo*	WS_2.3	
scoreProfile*	WS_2.3	
scoreThreshold*	WS_2.3	
scoreValue*	WS_2.3	
statementReference	WS_2.3	
transactionAmount	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.3	
transactionStatus	WS_2.3	
walletType	WS_2.3	

Table 36: Fields for paymentProviderFinalize response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.11 PAYMENTPROVIDERFINALIZE EXAMPLE

6.1.11.1 Example of paymentProviderFinalize request

```
{"interfaceVersion" : "IR_WS_2.0","keyVersion" : "1","merchantId" :
"011223344550000","messageVersion" : "0.1","redirectionData" : " RSs7cZyL1DqsVso
","transactionReference" : "TREFEXA2012","seal" :
"30218c85ca303d33d5e7b60bb48222b9826fbc30b6c2ec98fff285702e445ae1"}
```

6.1.11.1.1 Example of paymentProviderFinalize response

```
{"responseCode":"00","captureDay":"0","currencyCode":"978","merchantId":"011223344550000","pay
mentPattern":"ONE_SHOT","transactionAmount":"200","transactionDateTime":"2014-01-
14T08:38:11.494+01:00","transactionReference":"TREFEXA2012","seal":"79cf30535e08b8c6c9cfdd55c3
43845947f8fea892ec0ea1db96378055f71de6"}
```

6.1.12 CREDITTRANSFERINQUIRE SERVICE

Requests to retrieve the list of issuer's bank available:

This service is currently available for the payment means iDEAL.

This request provides a list of issuer's banks. If the request is successful, the merchant will receive in response a list of bank with their name and code. This information must be use for the initialization step (CreditTransferInitialize).

Requests include the following elements:

- Default inputs
- s10TransactionReference inputs

6.1.12.1 Default inputs for the creditTransferInquire

Field	Presence	As of version	Comments
currencyCode	Mandatory	WS_2.9	
customerLanguage	Optional	WS_2.9	
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
paymentMeanBrand	Mandatory	WS_2.9	
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.9	

Table 37: Fields for creditTransferInquire request

6.1.12.2 creditTransferInquire outputs

Responses include the following elements:

- Default outputs
- customerBank outputs

Field	As of version	Comments
customerBankList	WS_2.9	
errorFieldName	WS_2.9	
responseCode	WS_2.9	

Table 38: Fields for creditTransferInquire response

6.1.12.3 customerBank outputs

Field	Presence	As of version	Comments
customerBankCode	Optional	WS_2.9	
customerBankName	Optional	WS_2.9	

Table 39: Fields for the customerBank element

6.1.13 CREDITTRANSFERINQUIRE EXAMPLE

6.1.13.1 Example of creditTransferInquire request

```
{"currencyCode" : "978","customerLanguage" : "fr","interfaceVersion" :
"IR_WS_2.9","keyVersion" : "1","merchantId" : "011223344550000","paymentMeanBrand" :
"IDEAL","transactionReference" : " TREFEXA2015","seal" :
"ec7be3e2a0086a86f66f74bb6e10f244e51e4e648c7dbbf7c89ce9aa78a569cf"}
```

6.1.13.2 Example of creditTransferInquire response

```
{"customerBankList":[{"customerBankCode":"RABONL2U","customerBankName":"BANK -
RABO"},{"customerBankCode":"INGBNL2A","customerBankName":"BANK -
ING"}],"responseCode":"00","seal":"c46465d780517ce593b4bd5afe789fe58d2bc972937aaa35707ec866826
5e6dc"}
```

6.1.14 CREDITTRANSFERINITIALIZE SERVICE

Requests for a bank transfer initialization:

This service is currently available for the payment means iDEAL.

This request initializes a session for a bank transfer. If the initialization step is successful, the merchant will receive in response a secure URL (redirectionUrl) to which the customer should be redirected to continue the bank transfer. This redirection must be made via a POST form.

In the request, the merchant shall also indicate the URL (merchantReturnUrl) to which the customer will be redirected to at the end of the external bank transfer. The merchant must then call the creditTransferrFinalizeAndOrder service to finalize the transaction.

Requests include the following elements:

- Default inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs

- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- holderAddress inputs
- holderContact inputs
- s10TransactionReference inputs
- deliveryData inputs

6.1.14.1 Default inputs for the creditTransferInitialize

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.9	
customerBankCode	Mandatory	WS_2.9	
customerBankName	Mandatory	WS_2.9	
currencyCode	Mandatory	WS_2.9	
customerId	Optional	WS_2.9	
customerIpAddress	Optional	WS_2.9	
customerLanguage	Optional	WS_2.9	
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
invoiceReference	Optional	WS_2.9	
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
merchantReturnUrl	Mandatory	WS_2.9	
merchantTransactionDateTime	Optional	WS_2.9	
orderChannel	Mandatory	WS_2.9	
orderId	Optional	WS_2.9	
paymentMeanBrand	Mandatory	WS_2.9	
paymentPattern	Optional	WS_2.9	
returnContext	Optional	WS_2.9	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.9	
transactionOrigin	Optional	WS_2.9	
transactionReference	Optional	WS_2.9	
billingAddress	Optional	WS_2.9	See cardOrder
billingContact	Optional	WS_2.9	See cardOrder
customerAddress	Optional	WS_2.9	See cardOrder
customerContact	Optional	WS_2.9	See cardOrder
deliveryAddress	Optional	WS_2.9	See cardOrder
deliveryContact	Optional	WS_2.9	See cardOrder
fraudData	Optional	WS_2.10	See cardOrder
holderAddress	Optional	WS_2.9	See cardOrder
holderContact	Optional	WS_2.9	See cardOrder
customerData	Optional	WS_2.9	See cardOrder

Field	Presence	As of version	Comments
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2

Table 40: Fields for creditTransferInitialize request

6.1.14.2 creditTransferInitialize outputs

Field	As of version	Comments
errorFieldName	WS_2.9	
responseCode	WS_2.9	
redirectionUrl	WS_2.9	
redirectionData	WS_2.9	
messageVersion	WS_2.9	

Table 41: Fields for creditTransferInitialize response

6.1.15 CREDITTRANSFERINITIALIZE EXAMPLE

6.1.15.1 Example of creditTransferInitialize request

```
{"amount" : "1000","currencyCode" : "978","customerBankCode" : "RABONL2U","customerBankName"
: "BANK - RABO","customerLanguage" : "fr","interfaceVersion" : "IR_WS_2.9","keyVersion" :
"1","merchantId" : "011223344550000","merchantReturnUrl" :
"http://www.yoursiteweb.com","merchantTransactionDateTime" : "2015-05-
28T11:07:11.005+02:00","orderChannel" : "INTERNET","orderId" : "123","paymentMeanBrand" :
"IDEAL","transactionReference" : " TREFEXA2015","seal" :
"50e15601a2e83bcc5ffe596a51bab5b8dab2e3c62d3de2fe0d15502600f4aded"}
```

6.1.15.2 Example of creditTransferInitialize response

```
{"responseCode":"00","redirectionUrl":"https://payment-web.sips-
atos.com/paymentprovider/init","redirectionData":"RSs7cZyL1D...a+hZ+1ticHD/4/s=","messageVersi
on":"0.1","seal":"d0814427cbd5515e78cb54476c8594eb2c1d3732a9b392afd325ebae33945069"}
```

6.1.16 POST FORM FROM THE EXTERNAL BANK TRANSFER SERVICE

Once the bank transfer is processed, the customer is redirected to the merchant website. This redirection sends a POST form to the URL indicated in merchantReturnUrl at the creditTransferInitialize request.

The merchant must retrieve the fields and send them to the creditTransferFinalizeAndOrder request.

6.1.16.1 POST form outputs

Field	As of version	Comments
amount	WS_2.9	
merchantId	WS_2.9	
messageVersion	WS_2.9	

Field	As of version	Comments
redirectionData	WS_2.9	
responseCode	WS_2.9	
transactionDate	WS_2.9	
transactionId	WS_2.9	
transactionReference	WS_2_9	

Table 42: Fields for POST form response

6.1.17 CREDITTRANSFERFINALIZEANDORDER SERVICE

Requests for a bank transfer finalization (iDEAL):

This service is currently available for the payment means iDEAL.

This request is mandatory to know the result of the bank transfer. It must be called after the merchant received the POST form from the bank transfer service through the merchantReturnUrl (see creditTransferInitialize request).

Requests include the following elements:

- Default inputs
- s10TransactionReference inputs

6.1.17.1 Default inputs for the creditTransferFinalizeAndOrder

Field	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
messageVersion	Mandatory	WS_2.9	
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.9	
redirectionData	Mandatory	WS_2.9	

Table 43: Fields for creditTransferFinalizeAndOrder request

6.1.17.2 creditTransferFinalizeAndOrder outputs

Responses include the following elements:

- Default outputs

- s10TransactionReference outputs

	As of version	Comments
amount	WS_2.9	
authorisationId	WS_2.9	
captureDay	WS_2.9	
captureMode	WS_2.9	
complementaryCode	WS_2.9	
complementaryInfo	WS_2.9	
currencyCode	WS_2.9	
customerBankCode	WS_2.9	
customerBankName	WS_2.9	
customerId	WS_2.9	
customerIpAddress	WS_2.9	
errorFieldName	WS_2.9	
holderAuthentMethod*	WS_2.9	
holderAuthentProgram*	WS_2.9	
holderAuthentStatus*	WS_2.9	
invoiceReference	WS_2.9	
merchantId	WS_2.9	
merchantTransactionDateTime	WS_2.9	
orderChannel	WS_2.9	
orderId	WS_2.9	
paymentMeanBrand	WS_2.9	
paymentPattern	WS_2.9	
responseCode	WS_2.9	
returnContext	WS_2.9	
s10TransactionReference	WS_2.9	See 6.1.1.2
scoreColor*	WS_2.9	
scoreInfo*	WS_2.9	
scoreProfile*	WS_2.9	
scoreThreshold*	WS_2.9	
scoreValue*	WS_2.9	
statementReference	WS_2.9	
transactionDateTime	WS_2.9	
transactionReference	WS_2.9	

Table 44: Fields for creditTransferFinalizeAndOrder response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.18 CREDITTRANSFERFINALIZEANDORDER EXAMPLE

6.1.18.1 Example of creditTransferFinalizeAndOrder request

```
{"interfaceVersion" : "IR_WS_2.6","keyVersion" : "1","merchantId" :
"011223344550000","messageVersion" : "0.1","redirectionData" :
"RSs7cZyL1Dq....MzMbzqw","transactionReference" : "TREFEXA2015","seal" :
"f8769766243dcb603e417556a0e1d54b29b64cb9398317af3a16eb0a19db7057"}
```

6.1.18.2 Example of creditTransferFinalizeAndOrder response

```
{"amount":1000,"customerBankCode":"RABONL2U","customerBankName":"BANK -
RABO","captureDay":0,"captureMode":"AUTHOR_CAPTURE","currencyCode":"978","merchantTransactionD
ateTime":"2014-12-
05T00:00:00+01:00","merchantId":"011223344550000","orderChannel":"INTERNET","orderId":"123","p
aymentMeanBrand":"IDEAL","paymentPattern":"ONE_SHOT","responseCode":"00","s10TransactionRefere
nce":{"s10TransactionId":"129192","s10TransactionIdDate":"20150528"},"transactionDateTime":"20
15-05-
28T11:18:26+02:00","transactionReference":"TREFEXA2015","seal":"85cd304554ebc24a60255dbb695b34
c46665c6ad2f838966941ceb6dacd9e6ec"}
```

6.1.19 CARDCHECKENROLLMENT SERVICE

Requests for payment initialization via card with 3D-Secure process:

In this document, unless otherwise stated, any reference to 3-D Secure includes Visa (Verified by Visa), MasterCard (SecureCode) and American Express (SafeKey).

This request initializes a transaction on Sips platform and checks the card enrollment. If the enrollment step is successful, the merchant will receive in response a secure URL (redirectionUrl) to which the customer should be redirected to continue the authentication. This redirection must be made via a POST form.

Requests include the following elements:

- Default inputs
- authenticationData inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- s10TransactionReference inputs
- deliveryData inputs

6.1.19.1 Default inputs for the cardCheckEnrollment

Field	Presence	As of version	Comments

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.9	
captureDay	Optional	WS_2.9	
captureMode	Optional	WS_2.9	
cardCSCValue	Optional	WS_2.9	
cardExpiryDate	Optional	WS_2.9	
cardNumber	Mandatory	WS_2.9	
currencyCode	Mandatory	WS_2.9	
customerId	Optional	WS_2.9	
customerIpAddress	Optional	WS_2.9	
customerLanguage	Optional	WS_2.9	
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
invoiceReference	Optional	WS_2.9	
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
merchantName	Optional	WS_2.9	
merchantReturnUrl	Optional	WS_2.9	
merchantTransactionDateTime	Optional	WS_2.9	
merchantUrl	Optional	WS_2.9	
orderChannel	Mandatory	WS_2.9	
orderId	Optional	WS_2.9	
panType	Optional	WS_2.11	
paymentPattern	Optional	WS_2.9	
returnContext	Optional	WS_2.9	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.9	
transactionOrigin	Optional	WS_2.9	
transactionReference	Optional	WS_2.9	
authenticationData	Optional	WS_2.9	See cardOrder
billingAddress	Optional	WS_2.9	See cardOrder
billingContact	Optional	WS_2.9	See cardOrder
customerAddress	Optional	WS_2.9	See cardOrder
customerContact	Optional	WS_2.9	See cardOrder
deliveryAddress	Optional	WS_2.9	See cardOrder
deliveryContact	Optional	WS_2.9	See cardOrder
fraudData	Optional	WS_2.9	See cardOrder
holderAddress	Optional	WS_2.9	See cardOrder
holderContact	Optional	WS_2.9	See cardOrder
customerData	Optional	WS_2.9	See cardOrder
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2

Table 45: Fields for cardCheckEnrollment request

6.1.19.2 cardCheckEnrollment outputs

Responses include the following elements:

- Default outputs

Field	As of version	Comments
errorFieldName	WS_2.9	
messageVersion	WS_2.9	
paReqMessage	WS_2.9	
redirectionData	WS_2.9	
redirectionStatusCode	WS_2.9	
redirectionUrl	WS_2.9	

Table 46: Fields for cardCheckEnrollment response

6.1.20 CARDCHECKENROLLMENT EXAMPLE

6.1.20.1 Example of cardCheckEnrollment request

```
{"amount" : "1000","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"123","cardExpiryDate" : "201602","cardNumber" : "1234123412341234","currencyCode" :
"978","interfaceVersion" : "IR_WS_2.9","keyVersion" : "1","merchantId" :
"011223344550000","merchantTransactionDateTime" : "2015-05-
28T11:23:14.211+02:00","orderChannel" : "INTERNET","orderId" : "1234","transactionReference"
: "TREFEXA2015","seal" : "607083393a51b4b23261270df83c28a61e06f74928bc1e868c0abb9325dde2f2"}
```

6.1.20.2 Example of cardCheckEnrollment response

```
{"redirectionUrl":"http://www.acssiteweb.com","paReqMessage":"eJxVUU1zgj...Rv8P/AJQjttc=","red
irectionData":"uqjeV+Keg...MAwXk2jDJAe2TI=","redirectionStatusCode":"00","messageVersion":"0.1
","seal":"3c40cf8c12a8eebdfc114c83db8f944b77932b03814d8bf43ea792c20d05e7a6"}
```

6.1.21 WALLETCHECKENROLLMENT SERVICE

Requests for payment initialization via electronic wallet with 3D-Secure process:

In this document, unless otherwise stated, any reference to 3-D Secure includes Visa (Verified by Visa), MasterCard (SecureCode) and American Express (SafeKey).

This request initializes a transaction on Sips platform and checks the card enrollment retrieved from the electronic wallet. If the enrollment step is successful, the merchant will receive in response a secure URL (redirectionUrl) to which the customer should be redirected to continue the authentication. This redirection must be made via a POST form.

Requests include the following elements:

- Default inputs
- authenticationData inputs
- billingAddress inputs
- billingContact inputs
- customerAddress inputs
- customerContact inputs
- customerData inputs
- deliveryAddress inputs
- deliveryContact inputs
- fraudData inputs
- s10TransactionReference inputs
- deliveryData inputs

6.1.21.1 Default inputs for the walletCheckEnrollment

Field	Presence	As of version	Comments
amount	Mandatory	WS_2.9	
captureDay	Optional	WS_2.9	
captureMode	Optional	WS_2.9	
cardCSCValue	Optional	WS_2.9	
currencyCode	Mandatory	WS_2.9	
customerId	Optional	WS_2.9	
customerIpAddress	Optional	WS_2.9	
customerLanguage	Optional	WS_2.9	
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
invoiceReference	Optional	WS_2.9	
merchantId	Mandatory	WS_2.9	
merchantName	Optional	WS_2.9	
merchantReturnUrl	Optional	WS_2.9	
merchantTransactionDateTime	Optional	WS_2.9	
merchantUrl	Optional	WS_2.9	
merchantWalletId	Mandatory	WS_2.9	
orderChannel	Mandatory	WS_2.9	
orderId	Optional	WS_2.9	
paymentMeanId	Mandatory	WS_2.9	
paymentPattern	Optional	WS_2.9	
returnContext	Optional	WS_2.9	
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.9	
transactionOrigin	Optional	WS_2.9	
transactionReference	Optional	WS_2.9	
authenticationData	Optional	WS_2.9	See cardOrder
billingAddress	Optional	WS_2.9	See cardOrder
billingContact	Optional	WS_2.9	See cardOrder

Field	Presence	As of version	Comments
customerAddress	Optional	WS_2.9	See cardOrder
customerContact	Optional	WS_2.9	See cardOrder
deliveryAddress	Optional	WS_2.9	See cardOrder
deliveryContact	Optional	WS_2.9	See cardOrder
fraudData	Optional	WS_2.9	See cardOrder
holderAddress	Optional	WS_2.9	See cardOrder
holderContact	Optional	WS_2.9	See cardOrder
customerData	Optional	WS_2.9	See cardOrder
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2

Table 47: Fields for walletCheckEnrollment request

6.1.21.2 walletCheckEnrollment outputs

Responses include the following elements:

- Default outputs

Field	As of version	Comments
errorFieldName	WS_2.9	
messageVersion	WS_2.9	
paReqMessage	WS_2.9	
redirectionData	WS_2.9	
redirectionStatusCode	WS_2.9	
redirectionUrl	WS_2.9	

Table 48: Fields for walletCheckEnrollment response

6.1.22 WALLETCHECKENROLLMENT EXAMPLE

6.1.22.1 Example of walletCheckEnrollment request

```
{"amount" : "1000","captureDay" : "0","captureMode" : "AUTHOR_CAPTURE","cardCSCValue" :
"123","currencyCode" : "978","interfaceVersion" : "IR_WS_2.9","keyVersion" : "1","merchantId"
: "011223344550000","merchantTransactionDateTime" : "2015-05-
28T11:30:14.232+02:00","merchantWalletId" : "WALLET01","orderChannel" : "INTERNET","orderId"
: "1234","paymentMeanId" : "10","transactionReference" : "TREFEXA2015","seal" :
"3bfc4e958ba4ea43090db3c0c0bdba7162c92727ca1338b2111b28ddd91056a1"}
```

6.1.22.2 Example of walletCheckEnrollment response

```
{"redirectionUrl":"http://www.acssiteweb.com","paReqMessage":"eJxVUU1zgj...Rv8P/AJQjttc=","red
irectionData":"uqjeV+Keg...MAwXk2jDJAe2TI=","redirectionStatusCode":"00","messageVersion":"0.1
","seal":"3c40cf8c12a8eebdfc114c83db8f944b77932b03814d8bf43ea792c20d05e7a6"}
```

6.1.23 POST FORM TO THE ACS

The merchant website must cause the cardholder browser to send the form containing the **PAReq** message to the ACS URL that was received in the redirectionUrl field of the cardCheckEnrollment or walletCheckEnrollment service. This action must be initiated through the cardholder browser.
As per VISA and Mastercard 3DSecure implementation guidelines, this action must be performed with as little action on the cardholder's part as possible.

6.1.23.1 Building the redirection form

The merchant website needs to build a PAReq form to redirect the end-user to the ACS.
That form will need to contain the following values:

- the **PAReq** message: paReqMessage returned by the cardCheckEnrollment or walletCheckEnrollment service.
- the **TermUrl**: the Merchant URL to which the final reply must be posted following authentication of the end-user. This shall be a merchant website URL to redirect the end-user to.
- the **MD** ("Merchant Data") field: containing merchant state data that must be returned to the merchant. This field shall be used to retrieve the session on the merchant website.
- The form must contain Content-Type and Content-Length headers.

The use of javascript is recommended for that action, but such implementations must also support a fall-back for environments that do not support it, as illustrated in the HTML example below.

6.1.23.2 Example of HTTP POST redirection code - PAReq

```
<html>
<head>
  <title>Title for Page</title>
</head>
<body OnLoad="OnLoadEvent();" >
  <form name="downloadForm" action="ACS URL HERE(redirectionUrl)" method="POST">
  <noscript>
  <br>
  <br>
  <center>
    <h1>Processing your 3-D Secure Transaction</h1>
    <h2>JavaScript is currently disabled or is not supported by your browser.<br></h2>
    <h3>Please click Submit to continue the processing of your 3-D Secure transaction.</h3>
    <input type="submit" value="Submit">
  </center>
  </noscript>
  <input type="hidden" name="PaReq" value="BASE-64 ENCODED PAREQ HERE(redirectionData)">
  <input type="hidden" name="TermUrl" value="TERM URL HERE">
  <input type="hidden" name="MD" value="MERCHANT DATA HERE">
  </form>
<SCRIPT LANGUAGE="Javascript" >
<!--
function OnLoadEvent()
{
document.downloadForm.submit();
}
//-->
</SCRIPT>
</body>
</html>
```

6.1.23.3 End-user authentication

The end-user authentication is handled directly on the ACS server. As a result the merchant can't have any impact on the authentication or the look and feel of these pages.

6.1.23.4 End-user redirection to the merchant

At the end of the authentication process the end-user is redirected to the merchant through an HTTP redirect.
The ACS will POST the authentication results as variables on that URL. The POST variables will be as follows:

Field	Comments
PARes	Must be provided into the CardValidateAuthentication service
MD	Merchant Data: containing merchant state data that must be returned to the merchant. This field shall be used to retrieve the session on the merchant website.

Table 49: Fields for POST form response

6.1.24 CARDVALIDATEAUTHENTICATIONANDORDER SERVICE

Requests for payment finalization and order with a 3D-Secure process:

In this document, unless otherwise stated, any reference to 3-D Secure includes Visa (Verified by Visa), MasterCard (SecureCode) and American Express (SafeKey).

This request is mandatory to perform the payment order with a 3D-Secure context (PARes message). It must be called after the merchant received the POST form from the ACS (Access Control Server) (see See Post form to the ACS).

Requests include the following elements:

- Default inputs
- s10TransactionReference inputs

6.1.24.1 Default inputs for the cardValidateAuthenticationAndOrder

Field	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9

Field	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
messageVersion	Mandatory	WS_2.9	
paResMessage	Mandatory	WS_2.9	
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.9	
redirectionData	Mandatory	WS_2.9	

Table 50: Fields for cardValidateAuthenticationAndOrder request

6.1.24.2 cardValidateAuthenticationAndOrder outputs

Responses include the following elements:

- Default outputs
- s10TransactionReference outputs

	As of version	Comments
acquirerResponseCode	WS_2.9	
acquirerResponseMessage	WS_2.9	
amount	WS_2.9	
authorisationId	WS_2.9	
avsAddressResponseCode	WS_2.9	
avsPostcodeResponseCode	WS_2.9	
captureDay	WS_2.9	
captureMode	WS_2.9	
cardCSCResultCode	WS_2.9	
cardExpiryDate	WS_2.9	
complementaryCode	WS_2.9	
complementaryInfo	WS_2.9	
currencyCode	WS_2.9	
customerId	WS_2.9	
customerIpAddress	WS_2.9	
errorFieldName	WS_2.9	
guaranteeIndicator *	WS_2.9	
holderAuthentMethod*	WS_2.9	
holderAuthentProgram*	WS_2.9	
holderAuthentRelegation*	WS_2.9	
holderAuthentProgram*	WS_2.9	
holderAuthentResponseCode*	WS_2.9	

	As of version	Comments
holderAuthentStatus*	WS_2.9	
invoiceReference	WS_2.9	
maskedPan	WS_2.9	
merchantId	WS_2.9	
merchantTransactionDateTime	WS_2.9	
orderChannel	WS_2.9	
orderId	WS_2.9	
panEntryMode	WS_2.9	
paymentMeanBrand	WS_2.9	
paymentPattern	WS_2.9	
responseCode	WS_2.9	
returnContext	WS_2.9	
s10TransactionReference	WS_2.9	See 6.1.1.2
scoreColor*	WS_2.9	
scoreInfo*	WS_2.9	
scoreProfile*	WS_2.9	
scoreThreshold*	WS_2.9	
scoreValue*	WS_2.9	
statementReference	WS_2.9	
tokenPan	WS_2.9	
transactionDateTime	WS_2.9	
transactionReference	WS_2.9	

Table 51: Fields for cardValidateAuthenticationAndOrder response

*: these fields are provided when they are available, depending on the status of the transaction and the method of payment selected.

6.1.25 CARDVALIDATEAUTHENTICATIONANDORDER EXAMPLE

6.1.25.1 Example of cardValidateAuthenticationAndOrder request

```
{"interfaceVersion" : "IR_WS_2.9","keyVersion" : "1","merchantId" :
"011223344550000","messageVersion" : "0.1","paResMessage" :
"eJydVV...0t+v+W/v/D/BbSqpU8=","redirectionData" :
"uqjeV+KegCSM0POI...CjVUfxqyFq7zSYus7E=","transactionReference" : "TREFEXA2015","seal" :
"9565b71cb583f025278de91940c69900dee72d91e8184ddb6547f00927597fd1"}
```

6.1.25.2 Example of cardValidateAuthenticationAndOrder response

```
{"amount":1000,"authorisationId":"010628","avsAddressResponseCode":"1","avsPostcodeResponseCod
e":"1","captureDay":0,"captureMode":"AUTHOR_CAPTURE","cardExpiryDate":"201602","currencyCode":
"978","holderAuthentStatus":"3D_SUCCESS","guaranteeIndicator":"Y","maskedPan":"4907##########0
0","merchantId":"011223344550000","orderChannel":"INTERNET","orderId":"1234","paymentMeanBrand
":"VISA","responseCode":"00","transactionDateTime":"2015-05-
28T12:38:25+03:00","transactionReference":"TREFEXA2015","s10TransactionReference":{"s10Transac
tionId":"129306","s10TransactionIdDate":"20150528"},"seal":"ef3f3cc68a39a5d1144663162a35fa6070
bfe39d5abbe672b15417ff87afc431"}
```

6.1.26 CARDVALIDATEAUTHENTICATION SERVICE

Requests for 3D-Secure validation of authentication:

In this document, unless otherwise stated, any reference to 3-D Secure includes Visa (Verified by Visa), MasterCard (SecureCode) and American Express (SafeKey).

This request is mandatory to check the complete 3D-Secure authentication (with PARes message). It must be called after the merchant received the POST form from the ACS (Access Control Server) (see See Post form to the ACS).

Requests include the following elements:

- Default inputs
- s10TransactionReference inputs

6.1.26.1 Default inputs for the cardValidateAuthentication

Field	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.9	Value fixed at IR_WS_2.9
keyVersion	Mandatory	WS_2.9	
merchantId	Mandatory	WS_2.9	
messageVersion	Mandatory	WS_2.9	
paResMessage	Mandatory	WS_2.9	
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.9	
redirectionData	Mandatory	WS_2.9	

Table 52: Fields for cardValidateAuthentication request

6.1.26.2 cardValidateAuthentication outputs

Responses include the following elements:

- Default outputs

	As of version	Comments
responseCode	WS_2.9	
holderAuthentResponseCode	WS_2.9	

Table 53: Fields for cardValidateAuthentication response

6.1.27 CARDVALIDATEAUTHENTICATION EXAMPLE

6.1.27.1 Example of cardValidateAuthentication request

```
{"interfaceVersion" : "IR_WS_2.9","keyVersion" : "1","merchantId" :
"011223344550000","messageVersion" : "0.1","paResMessage" :
"eJydVV...0t+v+W/v/D/BbSqpU8=","redirectionData" :
"uqjeV+KegCSM0POI...CjVUfxqyFq7zSYus7E=","transactionReference" : "TREFEXA2015","seal" :
"9565b71cb583f025278de91940c69900dee72d91e8184ddb6547f00927597fd1"}
```

6.1.27.2 Example of cardValidateAuthentication response

```
{"responseCode":"00","holderAuthentResponseCode":"00,"seal":"ef3f3cc68a39a5d1144663162a35fa607
0bfe39d5abbe672b15417ff87afc431"}
```

6.2 DESCRIPTION OF THE CASH MANAGEMENT WEB SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement`

This service allows you to complete operations on existing transactions. For example, cancel and refund operations are not independent, they must be applied to a transaction.

6.2.1 ACCEPT CHALLENGE SERVICE

This function allows a merchant to accept the challenge on a transaction. Once the challenge is validated, the transaction retakes its normal course.

- Default inputs
- s10TransactionReference inputs

6.2.1.1 Accept Challenge default inputs

Fields	Presence	As of version	Comments
comment	Optional	WS_2.6	
interfaceVersion	Mandatory	WS_2.6	Value fixed at CR_WS_2.9
merchantId	Mandatory	WS_2.6	
operationOrigin	Optional	WS_2.6	
s10TransactionReference	Optional	WS_2.6	
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	
validationIndicator	Optional	WS_2.6	Accepted values: “Y” and “N”

Table 54: Fields for the accept challenge operation

6.2.1.2 Accept Challenge outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.6	
authorisationId	WS_2.6	
captureLimitDate	WS_2.6	
newStatus	WS_2.6	
operationDateTime	WS_2.6	

responseCode	WS_2.6	

Table 55: Fields for accept challenge response

6.2.2 EXAMPLE OF AN ACCEPT CHALLENGE OPERATION

6.2.2.1 Example of an accept challenge request

```
{"comment" : "good id","interfaceVersion" : "CR_WS_2.6","keyVersion" : "1","merchantId" :
"011223344550000","operationOrigin" : "SIPS-SIM","transactionReference" :
"SIM20140121510382","validationIndicator" : "Y","seal" :
"e746314133f7aaabd8ac33db0daf941eda744e8ff70b700490792ffe3480087d"}
```

6.2.2.2 Example of an accept challenge response

```
{"acquirerResponseCode":”00”, "authorisationId":”123”, "captureLimitDate":”1”,
"operationDateTime":"2014-10-16T23:51:42-12:00","responseCode":"00","newStatus" :
"VALIDATED","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.3 CANCEL SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement/cancel`

This function makes it possible to edit the amount of a transaction or to cancel a transaction to be sent to a bank. Requests for cancellation operations include the following elements:

- Default inputs
- s10TransactionReference inputs

✍ **Note**

> Any cancellation operation sent between 10 pm and 11 pm (French local time) will automatically return a response code 24 and the operation will not be processed.

6.2.3.1 Cancellation default inputs

Fields	Presence	As of version	Comments
operationAmount	Mandatory	WS_2.3	
currencyCode	Mandatory	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
operationOrigin	Mandatory	WS_2.3	

Fields	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.3	Value fixed at CR_WS_2.3
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	

Table 56: Fields for the cancellation operation

6.2.3.2 Cancellation outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
newAmount	WS_2.3	
newStatus	WS_2.3	
operationDateTime	WS_2.3	
responseCode	WS_2.3	

Table 57: Fields for cancellation response

6.2.4 EXAMPLE OF A CANCELLATION OPERATION

6.2.4.1 Example of a cancellation request

```
{"currencyCode" : "978","interfaceVersion" : "CR_WS_2.3","merchantId" :
"011223344550000","operationAmount" : "120","operationOrigin" :
"SO_BATCH","transactionReference" : "TREFEXA2012","keyVersion" : "1","seal" :
"bed2f99d4137d402fde0c86a182f7d4f01c68e57d8a073fe5e920c57d06da1b5"}
```

6.2.4.2 Example of a cancellation response

```
{"newAmount":0,"operationDateTime":"2012-10-16T23:51:42-
12:00","responseCode":"00","newStatus" :
"CANCELLED","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.5 CREDITHOLDER SERVICE

This function allows the merchant who has the banking data of their clients to credit their account without any prior transaction.

Requests for credit holder operations include the following elements:

- Default inputs

- s10TransactionReference inputs

6.2.5.1 creditHolder default inputs

Fields	Presence	As of version	Comments
amount	Mandatory	WS_2.0	
cardEffectiveDate	Optional	WS_2.0	
cardExpiryDate	Mandatory	WS_2.0	
cardNumber	Mandatory	WS_2.0	
cardScheme	Optional	WS_2.0	
cardSeqNumber	Optional	WS_2.0	
currencyCode	Mandatory	WS_2.0	
customerEmail	Optional	WS_2.0	
customerId	Optional	WS_2.0	
customerIpAddress	Optional	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed at CR_WS_2.2
merchantId	Mandatory	WS_2.0	
orderChannel	Mandatory	WS_2.0	
orderId	Optional	WS_2.0	
returnContext	Optional	WS_2.0	
panType	Optional	WS_2.1	
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionOrigin	Optional	WS_2.0	
transactionReference	Optional	WS_2.6	

Table 58: Fields for the credit holder operation

6.2.5.2 credit holder outputs

Responses include the following elements:

- Default outputs
- s10TransactionReference outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.0	
authorisationId	WS_2.0	
maskedPan	WS_2.0	

Fields	As of version	Comments
newStatus	WS_2.0	
operationDateTime	WS_2.0	
responseCode	WS_2.0	
s10TransactionReference	WS_2.6	See 6.1.1.2
transactionReference	WS_2.6	

Table 59: Fields for credit holder response

6.2.6 EXAMPLE OF A CREDIT HOLDER OPERATION

6.2.6.1 Example of an credit holder request

```
{"amount" : "1000","cardExpiryDate" : "201508","cardNumber" :
"4907000000000600","currencyCode" : "978","customerEmail" : "toto@mail.com","customerId" :
"customerId1","customerIpAddress" : "127.0.0.1","interfaceVersion" : "CR_WS_2.6","keyVersion"
: "1","merchantId" : "011223344550000","orderChannel" : "INTERNET","orderId" :
"123","panType" : "PAN","returnContext" : "my return context","transactionOrigin" : "SIPS-
SIM","transactionReference" : "SIM2014012248339","seal" :
"5227610a0f14cc30872dffb60c2dae6a85c9e3f1e487ff29c307c2c05b93567d"}
```

6.2.6.2 Example of a credit holder response

```
{"maskedPan":"4907##########00","newStatus":"TO_CREDIT","operationDateTime":"2014-12-
24T09:34:09+02:00","responseCode":"00","s10TransactionReference":{"s10TransactionId":"5","s10T
ransactionIdDate":"20141224"},"transactionReference":"SIM2014012248339","seal":"cca287c6f011ea
942bb850fddc7eba495519a35cf0a73b7aab2b99800b5969bd"}
```

6.2.7 DUPLICATE SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/cashManagement/duplicate

This function makes it possible to create a new transaction using the bank details of an existing transaction.

Requests for duplication operations include the following elements:

- Default inputs
- deliveryAddress inputs
- deliveryContact inputs
- deliveryData inputs
- fraudData inputs
- s10TransactionReference inputs
- s10FromTransactionReference input
- shoppingCartDetail inputs

6.2.7.1 Duplicate default inputs

Fields	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
captureDay	Mandatory	WS_2.3	
captureMode	Mandatory	WS_2.3	
currencyCode	Mandatory	WS_2.3	
customerEmail	Optional	WS_2.3	
customerId	Optional	WS_2.3	
customerIpAddress	Optional	WS_2.3	
deliveryAddress	Optional	WS_2.1	See 6.1.1.7
deliveryContact	Optional	WS_2.1	See 6.1.1.8
deliveryData	Optional	WS_2.5	See 6.2.7.2
fromMerchantId	Mandatory	WS_2.3	
fromTransactionReference	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at CR_WS_2.9
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
merchantTransactionDateTime	Mandatory	WS_2.3	
orderChannel	Mandatory	WS_2.3	
orderId	Optional	WS_2.3	
returnContext	Mandatory	WS_2.3	
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
s10FromTransactionReference	Optional	WS_2.6	See 6.2.7.3
sealAlgorithm	Optional	WS_2.11	
shoppingCartDetail	Optional	WS_2.5	See 6.2.7.4
statementReference	Optional	WS_2.2	
transactionOrigin	Optional	WS_2.3	
transactionReference	Mandatory	WS_2.3	
fraudData	Optional	WS_2.3	See 6.1.1.9

Table 60: Fields for the duplicate operation

6.2.7.2 deliveryData inputs

Field	Presence	As of version	Comments
deliveryChargeAmount	Optional	WS_2.5	
deliveryMethod	Optional	WS_2.5	
deliveryMode	Optional	WS_2.5	
deliveryOperator	Optional	WS_2.5	
estimatedDeliveryDate	Optional	WS_2.5	

estimatedDeliveryDelay	Optional	WS_2.6	

Table 61: Fields for the deliveryData element

6.2.7.3 s10FromTransactionReference inputs

Field	Presence	As of version	Comments
s10FromTransactionId	Optional	WS_2.6	
S10FromTransactionIdDate	Optional	WS_2.6	

Table 62: Fields for the s10FromTransactionReference element

6.2.7.4 shoppingCartDetail inputs

Field	Presence	As of version	Comments
shoppingCartTotalAmount	Optional	WS_2.5	
shoppingCartTotalQuantity	Optional	WS_2.5	
shoppingCartTotalTaxAmount	Optional	WS_2.6	
mainProduct	Optional	WS_2.5	
List<shoppingCartItem>	Optional	WS_2.5	See 6.2.7.5

Table 63: Fields for the shoppingCartDetail element

6.2.7.5 shoppingCartItem inputs

Field	Presence	As of version	Comments
productName	Optional	WS_2.5	
productDescription	Optional	WS_2.5	
productCode	Optional	WS_2.5	
productSKU	Optional	WS_2.5	
productUnitAmount	Optional	WS_2.5	
productQuantity	Optional	WS_2.5	
productTaxRate	Optional	WS_2.5	
productUnitTaxAmount	Optional	WS_2.5	
productCategory	Optional	WS_2.5	

Table 64: Fields for the shoppingCartItem element

6.2.7.6 Duplicate outputs

Responses include the following elements:

- Default outputs
- cardData outputs
- s10TransactionReference outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
scoreColor*	WS_2.3	
cardData	WS_2.9	See 6.1.1.23
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
maskedPan	WS_2.1	
panExpiryDate	WS_2.1	
paymentMeanBrand	WS_2.1	
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreInfo*	WS_2.3	
scoreProfile*	WS_2.3	
scoreThreshold*	WS_2.3	
scoreValue	WS_2.3	
responseCode	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	

Table 65: Fields for duplicate response

*: These fields are provided when they are available, depending on the status of the transaction and the payment method selected.

6.2.8 EXAMPLE OF A DUPLICATE OPERATION

6.2.8.1 Example of a duplicate request

```
{"amount" : "1000","currencyCode" : "978","captureDay" : "0","captureMode" :
"AUTHOR_CAPTURE","customerEmail" : "customer@email.com","customerId" :
"654321","customerIpAddress" : "127.0.0.1","interfaceVersion" : " CR_WS_2.3","merchantId" :
"011223344550000","orderChannel" : "INTERNET","orderId" : "123456","returnContext" :
"ReturnContext","transactionOrigin" : "SO_BATCH","fromTransactionReference" :
"TREFEXA20122"","transactionReference" : "TREFEXA20121","keyVersion" : "1","seal" :
"db2bbdd9550b5b7dc316d4b2859a0c45189aeaaf46b899cd90d5bde272e6ae24"}
```

6.2.8.2 Example of a duplicate response

```
{"acquirerResponseCode":"00","authorisationId":"069308","responseCode":"00","transactionDateTi
me":"2012-10-16T23:48:14-
12:00","seal":"0fdd9a35319712a9feff14eee0f68bbd35bd8fb6000819b2d5cd17b2704d6b6f"}
```

6.2.9 REFERRAL SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement/referral`

This service makes it possible to force a transaction to referral.

Requests for forcing operations include the following elements:

- Default inputs

6.2.9.1 Referral default inputs

Fields	Presence	As of version	Comments
authorisationId	Mandatory	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
operationOrigin	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at CR_WS_2.9
sealAlgorithm	Optional	WS_2.11	
transactionReference	Mandatory	WS_2.3	

Table 66: Fields for the referral operation

6.2.9.2 Referral outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
newAmount	WS_2.3	
newStatus	WS_2.3	
operationDateTime	WS_2.3	
responseCode	WS_2.3	

Table 67: Fields for the referral response

6.2.10 EXAMPLE OF A REFERRAL OPERATION

6.2.10.1 Example of a referral request

```
{"authorisationId" : "112598470074","interfaceVersion" : "CR_WS_2.3","merchantId" :
"011223344550000","operationOrigin" : "SO_BATCH","transactionReference" :
"TREFEXA2012","keyVersion" : "1","seal" :
```

```
"2f0bff22f589a32b0f94037b1040da9d2006e70b8c969f8952c3203619a229ff"}
```

6.2.10.2 Example of a referral response

```
{"newAmount":0,"operationDateTime":"2012-10-16T23:45:34-
12:00","responseCode":"00","newStatus" :
"TO_CAPTURE","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.11 REFUND SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement/refund`

This function makes it possible to re-credit the buyer's account.

Requests for refund operations include the following elements:

- Default inputs
- paymentMeanData inputs
- s10TransactionReference inputs
- shoppingCartDetail inputs

6.2.11.1 Refund default inputs

Fields	Presence	As of version	Comments
operationAmount	Mandatory	WS_2.3	
currencyCode	Mandatory	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
operationOrigin	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at CR_WS_2.3
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
transactionReference	Optional	WS_2.6	
paymentMeanData	Optional	WS_2.5	See 6.2.11.2
sealAlgorithm	Optional	WS_2.11	
shoppingCartDetail	Optional	WS_2.11	See 6.2.7.4

Table 68: Fields for the refund operation

6.2.11.2 paymentMeanData inputs

Field	Presence	As of version	Comments
facilypay	Optional	WS_2.5	See 6.2.11.3

Table 69: Fields for the paymentMeanData element

6.2.11.3 facilypay inputs

Field	Presence	As of version	Comments
depositRefundIndicator	Optional	WS_2.5	

Table 70: Fields for the facilypay element

6.2.11.4 Refund outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
newAmount	WS_2.3	
newStatus	WS_2.3	
operationDateTime	WS_2.3	
responseCode	WS_2.3	

Table 71: Fields for refund response

6.2.12 EXAMPLE OF A REFUND OPERATION

6.2.12.1 Example of a refund request

```
{"currencyCode" : "978","interfaceVersion" : "CR_WS_2.3","merchantId" :
"011223344550000","operationAmount" : "1000","operationOrigin" :
"SO_BATCH","transactionReference" : "TREFEXA2012","keyVersion" : "1","seal" :
"bed2f99d4137d402fde0c86a182f7d4f01c68e57d8a073fe5e920c57d06da1b5"}
```

6.2.12.2 Example of a refund response

```
{"acquirerResponseCode":"00","authorisationId":"123456789102","newAmount":0,"operationDateTime
":"2012-10-16T23:45:34-12:00","responseCode":"00","newStatus" :
"TO_CREDIT","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.13 REFUSE CHALLENGE SERVICE

This function allows a merchant to refuse the challenge on a transaction. Once the challenge is refused, the transaction is refused.

- Default inputs
- s10TransactionReference inputs

6.2.13.1 Refuse Challenge default inputs

Fields	Presence	As of version	Comments
comment	Optional	WS_2.6	
interfaceVersion	Mandatory	WS_2.6	Value fixed at CR_WS_2.6
merchantId	Mandatory	WS_2.6	
operationOrigin	Optional	WS_2.6	
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	

Table 72: Fields for the refuse challenge operation

6.2.13.2 Refuse Challenge outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
newStatus	WS_2.6	
operationDateTime	WS_2.6	
responseCode	WS_2.6	

Table 73: Fields for refuse challenge response

6.2.14 EXAMPLE OF AN REFUSE CHALLENGE OPERATION

6.2.14.1 Example of an refuse challenge request

```
{"comment" : "bad id","interfaceVersion" : "CR_WS_2.6","keyVersion" : "1","merchantId" :
"011223344550000","operationOrigin" : "SIPS-SIM","transactionReference" :
"SIM20140121510382","seal" :
"c36bcfaf3c05ef257795149686c53022f212e05c6fac98d0169c5e9313941444"}
```

6.2.14.2 Example of an refuse challenge response

```
 {"operationDateTime":"2014-10-16T23:51:42-12:00","responseCode":"00","newStatus" :
"REFUSED","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.15 VALIDATION SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement/validate`

This function makes it possible to trigger the transmission of a transaction to a bank.

Requests for validation operations include the following elements:

- Default inputs
- s10TransactionReference inputs

6.2.15.1 Validation default inputs

Fields	Presence	As of version	Comments
operationAmount	Mandatory	WS_2.3	
currencyCode	Mandatory	WS_2.3	
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
operationOrigin	Optional	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at CR_WS_2.3
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	

Table 74: Fields for the validation operation

6.2.15.2 Validation outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
newAmount	WS_2.3	
newStatus	WS_2.3	
operationDateTime	WS_2.3	
responseCode	WS_2.3	

Table 75: Fields for the validation response

6.2.16 EXAMPLE OF A VALIDATION OPERATION

6.2.16.1 Example of a validation request

```
{"currencyCode" : "978","interfaceVersion" : "CR_WS_2.3","merchantId" :
"011223344550000","operationAmount" : "1200","operationOrigin" :
```

```
"SO_BATCH","transactionReference" : "TREFEXA2012","keyVersion" : "1","seal" :
"bed2f99d4137d402fde0c86a182f7d4f01c68e57d8a073fe5e920c57d06da1b5"}
```

6.2.16.2 Example of a validation response

```
{"acquirerResponseCode":"00","authorisationId":"123456789102","newAmount":1000,"operationDateT
ime":"2012-10-16T23:53:19-12:00","responseCode":"00","newStatus" :
"TO_CAPTURE","seal":"919b544bfb60539c8979992bc469e0b4229030ee2462a51f62c9cbd3510b3647"}
```

6.2.17 RECYCLE SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/cashManagement/recycle`

This function makes it possible to create a new transaction using the bank details of an existing transaction. This operation is similar to duplicate operation, but include some limitations.

Requests for recycle operations include the following elements:

- Default inputs
- s10TransactionReference inputs

6.2.17.1 recycle default inputs

Fields	Presence	As of version	Comments
amount	Mandatory	WS_2.3	
captureDay	Optional	WS_2.3	
captureMode	Optional	WS_2.3	
currencyCode	Mandatory	WS_2.3	
fromTransactionReference	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
merchantTransactionDateTime	Optional	WS_2.0	
s10FromTransactionReference	Optional	WS_2.6	See 6.2.7.3
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
statementReference	Optional	WS_2.0	
transactionOrigin	Optional	WS_2.0	
transactionReference	Optional	WS_2.6	

Table 76: Fields for the recycle operation

6.2.17.2 Recycle outputs

Responses include the following elements:

- Default outputs
- cardData outputs
- s10TransactionReference outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.3	
authorisationId	WS_2.3	
cardData	WS_2.9	See 6.1.1.23
complementaryCode	WS_2.3	
complementaryInfo	WS_2.3	
maskedPan	WS_2.3	
panExpiryDate	WS_2.3	
paymentMeanBrand	WS_2.3	
responseCode	WS_2.3	
s10TransactionReference	WS_2.6	See 6.1.1.2
scoreColor	WS_2.3	
scoreInfo	WS_2.3	
scoreProfile	WS_2.3	
scoreThreshold	WS_2.3	
scoreValue	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	

Table 77: Fields for the recycle response

6.2.18 EXAMPLE OF A RECYCLE OPERATION

6.2.18.1 Example of a recycle request

```
{"amount" : "1000","captureMode" : "AUTHOR_CAPTURE","currencyCode" :
"978","fromTransactionReference" : "SIM2014082813176","interfaceVersion" :
"CR_WS_2.0","merchantId" : "011223344550000","merchantTransactionDateTime" : "2014-08-
28T13:17:16.480+02:00","transactionOrigin" : "SIPS-SIM","transactionReference" :
"SIM20140828131716","seal" :
"bab5067f4fa23924cd80f35d5585519a563ff261f38dfa9b74fbb76888b00b64","keyVersion" : "1"}
```

6.2.18.2 Example of a recycle response

```
{"acquirerResponseCode":"00","authorisationId":"736256","maskedPan":"4541##########00","panExp
iryDate":"201504","paymentMeanBrand":"VISA","scoreValue":"-
2.3","scoreColor":"RED","scoreInfo":"A3;N;NOT_APPLICABLE#WN;P;U#BN;N;U#GN;N;U#SC;N;TRANS=3:2;C
UMUL=3000:250000","scoreProfile":"myprofSco","scoreThreshold":"-
3.0","responseCode":"00","transactionDateTime":"2014-08-
28T13:20:45+02:00","seal":"df0c23e0a60a745f1f546168fafedb7ae547912473d76c93ee52d31b9d714765"}
```

6.2.19 WALLETCREDITHOLDER SERVICE

This function allows the merchant who has the wallet data of their clients to credit their account without any prior transaction.

Requests for wallet credit holder operations include the following elements:

- Default inputs
- s10TransactionReference inputs

6.2.19.1 wallet creditHolder default inputs

Fields	Presence	As of version	Comments
amount	Mandatory	WS_2.9	
currencyCode	Mandatory	WS_2.9	
customerEmail	Optional	WS_2.9	
customerId	Optional	WS_2.9	
customerIpAddress	Optional	WS_2.9	
interfaceVersion	Mandatory	WS_2.9	Value fixed at CR_WS_2.9
merchantId	Mandatory	WS_2.9	
merchantWalletId	Mandatory	WS_2.9	
orderChannel	Mandatory	WS_2.9	
orderId	Optional	WS_2.9	
paymentMeanId	Mandatory	WS_2.9	
returnContext	Optional	WS_2.9	
panType	Optional	WS_2.9	
s10TransactionReference	Optional	WS_2.9	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionOrigin	Optional	WS_2.9	
transactionReference	Optional	WS_2.9	

Table 78: Fields for the wallet credit holder operation

6.2.19.2 wallet credit holder outputs

Responses include the following elements:

- Default outputs
- s10TransactionReference outputs

Fields	As of version	Comments
acquirerResponseCode	WS_2.9	
authorisationId	WS_2.9	
maskedPan	WS_2.9	
newStatus	WS_2.9	
operationDateTime	WS_2.9	
paymentMeanBrand	WS_2.9	
responseCode	WS_2.9	
s10TransactionReference	WS_2.9	See 6.1.1.2
transactionReference	WS_2.9	

Table 79: Fields for wallet credit holder response

6.2.20 EXAMPLE OF AWALLET CREDIT HOLDER OPERATION

6.2.20.1 Example of a wallet credit holder request

```
{"amount" : "1000","currencyCode" : "978","customerEmail" : "toto@mail.fr","customerId" :
"customerId1","customerIpAddress" : "127.0.0.1","interfaceVersion" : "CR_WS_2.6","keyVersion"
: "1","merchantId" : "011223344550000","merchantWalletId" : "SIM01","orderChannel" :
"INTERNET","orderId" : "123","paymentMeanId" : "2","returnContext" : "my return
context","transactionOrigin" : "SIPS-SIM","transactionReference" : "SIM2014012248339","seal"
: "d3f5f2c4c6f608e3a4940c6700ca6827c066715512be835660a8091c6ec1e098"}
```

6.2.20.2 Example of a wallet credit holder response

```
{"maskedPan":"4907##########00","newStatus":"TO_CREDIT","operationDateTime":"2015-05-
06T12:12:07+03:00","responseCode":"00","s10TransactionReference":{"s10TransactionId":"34800","
s10TransactionIdDate":"20150506"},"transactionReference":"SIM2014012248339","paymentMeanBrand"
:"VISA","seal":"250c2a5831ae3b1a015d4d8be0e1970936e2c53b50607c8f6182556c0291ab8d"}
```

6.3 Description of the diagnostic web service

The URL is `https://office-server.sips-atos.com/rs-services/v2/diagnostic`

This service allows you to request information on existing transactions.

6.3.1 GETTRANSACTIONDATA SERVICE

The URL is
`https://office-server.sips-atos.com/rs-services/v2/diagnostic/getTransactionData`

This function makes it possible to recover information regarding a specific transaction previously set up using Sips which is stored in the Sips database.

Requests for validation operations include the following elements:

- Default inputs
- s10TransactionReference inputs

6.3.1.1 getTransactionData default inputs

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed at DR_WS_2.9
s10TransactionReference	Optional	WS_2.6	See 6.1.1.2
sealAlgorithm	Optional	WS_2.11	
transactionReference	Optional	WS_2.6	

Table 80: Fields for the getTransactionData request

6.3.1.2 GetTransactionData outputs

Responses include the following elements:

- Default outputs
- cardData outputs
- s10FromTransactionReference outputs
- s10TransactionReference outputs

Fields	As of version	Comments

Fields	As of version	Comments
acquirerResponseMessage	WS_2.9	
authorisationId	WS_2.3	
automaticResponseStatus	WS_2.3	
captureLimitDate	WS_2.3	
cardData	WS_2.9	See 6.1.1.23
currencyCode	WS_2.3	
currentAmount	WS_2.3	
customerId	WS_2.3	
customerIpAddress	WS_2.3	
dccStatus	WS_2.3	
fromTransactionReference	WS_2.3	
holderAuthentMethod	WS_2.3	
holderAuthentProgram	WS_2.3	
holderAuthentStatus	WS_2.3	
lastOperationDateTime	WS_2.3	
lastOperationName	WS_2.3	
maskedPan	WS_2.3	
merchantId	WS_2.3	
orderChannel	WS_2.3	
orderId	WS_2.3	
originAmount	WS_2.3	
panEntryMode	WS_2.3	
panExpiryDate	WS_2.3	
paymentMeanBrand	WS_2.3	
paymentMeanType	WS_2.3	
paymentPattern	WS_2.3	
remainingAmount	WS_2.3	
responseCode	WS_2.3	
s10FromTransactionReference	WS_2.6	See 6.2.7.3
s10TransactionReference	WS_2.6	See 6.1.1.2
tokenPan	WS_2.3	
transactionDateTime	WS_2.3	
transactionReference	WS_2.6	
transactionStatus	WS_2.3	
walletType	WS_2.3	

Table 81: Fields for the getTransactionData response

6.3.2 EXAMPLE OF A GETTRANSACTIONDATA INFORMATION REQUEST

6.3.2.1 Example of a getTransactionData request

```
{"interfaceVersion" : "DR_WS_2.3","merchantId" : "011223344550000", "transactionReference" :
"TREFEXA2012","seal" : "112a4b079ece08a0a55511cd5469fc47051d6ddb1404623170ba3873668e5c58" }
```

6.3.2.2 Example of a getTransactionData response

```
{"automaticResponseStatus":"UNDEFINED","authorisationId":"298392","captureLimitDate":"20140108
","paymentMeanType":"CARD","paymentMeanBrand":"MASTERCARD","currencyCode":"826","currentAmount
":1000,"customerId":"customerId1","customerIpAddress":"127.0.0.1","lastOperationDateTime":"201
4-01-
07T12:50:10+01:00","lastOperationName":"TRANSACTION","orderId":"123","originAmount":1000,"tran
sactionDateTime":"2014-01-
07T12:50:01+01:00","responseCode":"00","transactionStatus":"TO_CAPTURE","orderChannel":"INTERN
ET","paymentPattern":"ONE_SHOT","merchantId":"011223344550000","seal":"d971d516610f68050a71068
c49ae778c37eb1c9b16f2ec90a3feadeb1bb60876"}
```

6.4 DESCRIPTION OF THE WALLET WEB SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/wallet`

This service allows the management of customer wallet. A wallet allows you to manage payment means per customer and allows this customer to pay through his wallet. This web service is used only for management purpose and not for payment. Refer to the web service checkout for the wallet payment part.

6.4.1 ADDCARD SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/wallet/addCard`

This function allows to create a "Wallet" account with a card.
If the account already exists, a 02 response code is returned.
If the creation succeeds, response code 00 is returned as well as a lot of information about the account and the associated card:

- 'Wallet' identifier
- creation date
- external identifier of the payment mean created for the 'Wallet'
- card number, partially hidden (only the first 4 and 2 last digits are clear).

Requests for creation include the following elements

- Default inputs

6.4.1.1 addCard default inputs

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed to WR_WS_2.9
cardNumber	Mandatory	WS_2.3	
cardExpiryDate	Mandatory	WS_2.3	
merchantWalletId	Mandatory	WS_2.3	
paymentMeanAlias	Optional	WS_2.3	Example : « my visa card»
sealAlgorithm	Optional	WS_2.11	

Table 82: Fields for the addCard request

6.4.1.2 addCard outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletActionDateTime	WS_2.0	Available if response code = 00
paymentMeanId	WS_2.0	Available if response code = 00
maskedPan	WS_2.0	Available if response code = 00
walletResponseCode	WS_2.0	
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 83: Fields for the addCard response

6.4.2 EXAMPLE OF A ADDCARD SERVICE

6.4.2.1 Example of a addCard request

```
{"cardExpiryDate" : "201405","cardNumber" : "5219000000000000","interfaceVersion" : "
WR_WS_2.3","keyVersion" : "1","merchantId" : "011223344550000","merchantWalletId" :
"iDWal1","paymentMeanAlias" : "myvisacard","seal" :
"4b7beed20ff443b3c05cc904bcd793ba6cace54b9ff669cf26d8576e267dc03c"}
```

6.4.2.2 Example of a addCard response

```
{"walletActionDateTime":"2014-03-19T23:15:03-
12:00","paymentMeanId":"13","maskedPan":"4977##########55","walletResponseCode":"00","seal":"a
6671feade95c57085939fe973e8455a5c7a81d465f78f1b94c22f8b29a2b751"}
```

6.4.3 ADDDIRECTDEBIT SERVICE

This function creates a "Wallet" account with a Direct Debit payment mean.
If the account already exists, a 02 response code is returned.
If the creation succeeds, response code 00 is returned as well as a lot of information about the account and the associated card:

- creation date
- external identifier of the payment mean created for the 'Wallet'
- iban, partially hidden (only the first 4 and 2 last digits are clear).

Requests for creation include the following elements

- Default inputs

6.4.3.1 Default inputs for the addDirectDebit

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed to WR_WS_2.3

Fields	Presence	As of version	Comments
merchantWalletId	Mandatory	WS_2.3	
paymentMeanBrand	Mandatory	WS_2.3	
mandateId	Optional	WS_2.3	
customerBankCode	Optional	WS_2.3	
customerAccount	Optional	WS_2.3	
sealAlgorithm	Optional	WS_2.11	
transactionActors	Optional	WS_2.3	BtoB / BtoC
paymentMeanAlias	Optional	WS_2.3	Example : « ma carte visa »

Table 84: Fields for the addDirectDebit request

6.4.3.2 addDirectDebit outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletActionDateTime	WS_2.0	Available if response code = 00
paymentMeanId	WS_2.0	Available if response code = 00
maskedPan	WS_2.0	Available if response code = 00
transactionActors	WS_2.0	Available if response code = 00
walletResponseCode	WS_2.0	
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 85: Fields for the addDirectDebit response

6.4.4 EXAMPLE OF A ADDDIRECTDEBIT SERVICE

6.4.4.1 Example of a addDirectDebit request

```
{"customerAccount" : "FR7630046001290029721519 5464","customerBankCode" :
"CEPAFRPP118","interfaceVersion" : "WR_WS_2.0","keyVersion" : "1","mandateId" :
"MANDATE0001","merchantId" : "011223344550000","merchantWalletId" : "1","paymentMeanAlias" :
"ABC bank account","paymentMeanBrand" : "SEPA_DIRECT_DEBIT","transactionActors" :
"BtoB","seal" : "8493c63cd279824b90bf39f542df205343392e08412345d4c2b9c856c42464ef9"}
```

6.4.4.2 Example of a addDirectDebit response

```
{"walletActionDateTime":"2014-07-28T21:14:13-
12:00","paymentMeanId":"8","maskedPan":"MAND#####01","transactionActors":"BtoB","walletRespons
eCode":"00","seal":"75cc0bd5f72a15fb38b0f609393437db61866f867fa7befeef9b3f496a9e449f"}
```

6.4.5 SIGNOFF SERVICE

This function allows to remove a "Wallet" as well as its related payment means. If the account does not exist, a 01 response code is returned. If the deletion works, 00 response code is returned and the date of removal.

Requests for deletion include the following elements

- input elements

6.4.5.1 Default inputs for the signOff

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed to WR_WS_2.9
merchantWalletId	Mandatory	WS_2.3	
sealAlgorithm	Optional	WS_2.11	

Table 86: Fields for the signoff request

6.4.5.2 signOff outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletActionDateTime	WS_2.0	Available if response code = 00
walletResponseCode	WS_2.0	
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 87: Fields for the signOff response

6.4.6 EXAMPLE OF A SIGNOFF WALLET DELETION

6.4.6.1 Example of a signoff request

```
{"interfaceVersion" : "WR_WS_2.3","keyVersion" : "1","merchantId" :
"011223344550000","merchantWalletId" : "SIM01","seal" :
"8217a6163368bee3b5baebc47a822d40327344ce578411055f95d6e5752d00f4"}
```

6.4.6.2 Example of a signoff response

```
{"walletActionDateTime":"2014-05-19T23:20:25-
12:00","walletResponseCode":"00","seal":"8ff8b4302b5eeda23bad2d556dd539c2b3991f8dcd4deb66e338d
f3a1f59ebe6"}
```

6.4.7 UPDATEPAYMENTMEAN SERVICE

This function allows a merchant to update one of the payment means of his 'wallet '. If the account or the card do not exist, a 01 response code is returned. If the update works, 00 response code is returned and the date of update.

Requests for updating a payment mean include the following elements:

- input elements

6.4.7.1 Default inputs for the updatePaymentMean

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.3	
merchantId	Mandatory	WS_2.3	
interfaceVersion	Mandatory	WS_2.3	Value fixed to WR_WS_2.9
merchantWalletId	Mandatory	WS_2.3	
paymentMeanId	Mandatory	WS_2.3	
paymentMeanAlias	Optional	WS_2.3	
sealAlgorithm	Optional	WS_2.11	

Table 88: Fields for the updatePaymentMean request

6.4.7.2 updatePaymentMean outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletActionDateTime	WS_2.0	Available if response code = 00
walletResponseCode	WS_2.0	
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 89: Fields for the updatePaymentMean response

6.4.8 EXAMPLE OF AN UPDATE OF A PAYMENT MEAN UPDATEPAYMENTMEAN

6.4.8.1 Example of an updatePaymentMean request

```
{"interfaceVersion" : " WR_WS_2.3","keyVersion" : "1","merchantId" :
"011223344550000","merchantWalletId" : "walId011","paymentMeanAlias" :
"myAlias","paymentMeanId" : "2","seal" :
"1e26289499d5f6104e6300291f272d7b994a111eecf8feedccf934560de26e5d"}
```

6.4.8.2 Example of an updatePaymentMean response

```
{"walletActionDateTime":"2014-05-19T23:19:54-
12:00","walletResponseCode":"00","seal":"4d40a4af0fe8ceaf1370ab5e434c6ba939973bcacf33f202c72c4
7a8ab622c19"}
```

6.4.9 DELETEPAYMENTMEAN SERVICE

This function allows a merchant to permanently delete one of the payment means of his 'wallet '.

If the account or the card does not exist, a 01 response code is returned. If the deletion works, 00 response code is returned and the date of removal.

Requests for deletion of a payment mean include the following elements:

- input elements

6.4.9.1 Default inputs for the deletePaymentMean

Fields	Presence	As of version	Comments
merchantId	Mandatory	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed to WR_WS_2.3
merchantWalletId	Mandatory	WS_2.0	
paymentMeanId	Mandatory	WS_2.0	
sealAlgorithm	Optional	WS_2.11	

Table 90: Fields for the deletePaymentMean request

6.4.9.2 deletePaymentMean outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletActionDateTime	WS_2.0	Available if response code = 00

walletResponseCode	WS_2.0	
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 91: Fields for the deletePaymentMean response

6.4.10 EXAMPLE OF PAYMENT MEAN DELETION DELETEPAYMENTMEAN

6.4.10.1 Example of a deletePaymentMean request

```
{"interfaceVersion" : "WR_WS_2.3","keyVersion" : "2","merchantId" :
"011223344550000","merchantWalletId" : "walId011","paymentMeanId" : "2","seal" :
"dc02b07a50eaae5007be184ede9a1b03150809ced6fca81c81f7a54929129b44"}
```

6.4.10.2 Example of a deletePaymentMean response

```
{"walletActionDateTime":"2014-05-19T23:22:08-
12:00","walletResponseCode":"00","seal":"e721a45f7decf017ba079dd0c25b1e7916c6c9e6c7e86d6cef37b
ec2cf42ba04"}
```

6.4.11 GETWALLETDATA SERVICE

This function allows to consult a "Wallet" and payment means associated to it.

If the account does not exist, a 01 response code is returned. If the request works, 00 response code is returned as well as information related to the payment means.

Request for consultation of a wallet include the following elements:

- input elements

6.4.11.1 Default inputs for the getWalletData

Fields	Presence	As of version	Comments
merchantId	Mandatory	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed to WR_WS_2.9
merchantWalletId	Mandatory	WS_2.0	
sealAlgorithm	Optional	WS_2.11	

Table 92: Fields for the getWalletData request

6.4.11.2 getWalletData output

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletCreationDateTime	WS_2.0	Available if response code = 00
walletLastActionDateTime	WS_2.0	Available if response code = 00
walletResponseCode	WS_2.0	
walletPaymentMeanDataList	WS_2.0	List of walletPaymentMeanData Container fields, see below
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Tableau 93: Fields for the getWalletData response

6.4.11.3 walletPaymentMeanData element

Fields	As of version	Comments
paymentMeanId	WS_2.0	
maskedPan	WS_2.0	
paymentMeanAlias	WS_2.0	
panExpiryDate	WS_2.0	
paymentMeanBrand	WS_2.0	
transactionActors	WS_2.0	

Tableau 94: Fields for the walletPaymentMeanData element

6.4.12 EXAMPLE OF WALLET READING GETWALLETDATA

6.4.12.1 Example of a getWalletData request

```
{"interfaceVersion" : " WR_WS_2.3","keyVersion" : "1","merchantId" :
"011223344550000","merchantWalletId" : " walId011","seal" :
"8217a6163368bee3b5baebc47a822d40327344ce578411055f95d6e5752d00f4"}
```

6.4.12.2 Example of a getWalletData response

```
{"walletCreationDateTime":"2013-12-23T05:17:26-12:00","walletLastActionDateTime":"2014-01-
19T23:16:00-
12:00","walletResponseCode":"00","walletPaymentMeanDataList":[{"paymentMeanId":"14","maskedPan
":"4977##########02","paymentMeanBrand":"SEPA_DIRECT_DEBIT"},{"paymentMeanId":"13","maskedPan"
:"4977##########55","paymentMeanAlias":"MySDD","panExpiryDate":"201501","paymentMeanBrand":"CB
"}],"seal":"4579cfc4044c29550327f9cba0be400129e95cb5b2639c6e301484930b4f9f94"}
```

6.4.13 SERVICE GETPAYMENTMEANDATA

This function allows to consult a "Wallet" and one of these payment means information.

If the account or payment mean does not exist, a 01 response code is returned. If the request works, 00 response code is returned as well as information related to the payment mean.

Request for consultation of a payment mean include the following elements:

- input elements

6.4.13.1 Default inputs for the getPaymentMeanData

Fields	Presence	As of version	Comments
merchantId	Mandatory	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed to WR_WS_2.3
merchantWalletId	Mandatory	WS_2.0	
paymentMeanId	Mandatory	WS_2.0	
sealAlgorithm	Optional	WS_2.11	

Table 95: Fields for the getPaymentMeanData request

6.4.13.2 getPaymentMeanData outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
walletResponseCode	WS_2.0	
walletPaymentMeanData	WS_2.0	Container field, see getWalletData
errorFieldName	WS_2.0	Available if walletResponseCode 12 or 30

Table 96: Fields for the getPaymentMeanData response

6.4.14 EXAMPLE OF PAYMENT MEAN CONSULTATION GETPAYMENTMEANDATA

6.4.14.1 Example of a getPaymentMeanData request

```
{"interfaceVersion" : "WR_WS_2.3","keyVersion" : "1","merchantId" :
```

```
"011223344550000","merchantWalletId" : " walId011","paymentMeanId" : "14","seal" :
"7ba62b5dc8583a636fa35ec8399025e67d5a6335de4e04e25e99d21209e0bd4e"}
```

6.4.14.2 Example of a getPaymentMeanData response

```
{"walletResponseCode":"00","walletPaymentMeanData":{"paymentMeanId":"14","maskedPan":"4977####
######02","paymentMeanBrand":"SEPA_DIRECT_DEBIT"},"seal":"17f797e25668662aa51d59dbbd51c7094410
f46326d15568f73f27621380c34a"}
```

6.5 DESCRIPTION OF THE PAYMENTMEANINFO WEB SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/paymentMeanInfo

This service allows you to consult information about a payment mean, currently available for cards.

6.5.1 GETCARDDATA SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/paymentMeanInfo/getCardData

This function allows consulting of the card information associated with a card number or a card IIN.

If the card number or the card IIN does not exist, a 05 response code is returned. If the request works, 00 response code is returned as well as information related to the card.

Request for consultation of a card number or a card IIN include the following elements:

- input elements

6.5.1.1 Default inputs for the getCardData

Fields	Presence	As of version	Comments
keyVersion	Mandatory	WS_2.5	
merchantId	Mandatory	WS_2.5	
interfaceVersion	Mandatory	WS_2.5	Value fixed to PMR_WS_2.5
cardNumber	Optional	WS_2.5	Must be set if cardIIN is not set
cardIIN	Optional	WS_2.5	Must be set if cardNumber is not set
sealAlgorithm	Optional	WS_2.11	

Table 97: Fields for the getCardData request

6.5.1.2 getCardData output

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
paymentMeanInfoResponseCode	WS_2.5	
cardDataList	WS_2.5	List of cardData Container fields, see below
errorFieldName	WS_2.5	Available if paymentMeanInfoResponseCode 12 or 30

Tableau 98: Fields for the getCardData response

6.5.1.3 cardData element

Fields	As of version	Comments
cardBrand	WS_2.5	
cardCorporateIndicator	WS_2.5	
cardEffectiveDateIndicator	WS_2.5	
cardProductCode	WS_2.5	
cardProductName	WS_2.5	
cardProductProfile	WS_2.5	
cardScheme	WS_2.5	
cardSchemeCode	WS_2.5	
cardSeqNumberIndicator	WS_2.5	
issuerCode	WS_2.5	
issuerCountryCode	WS_2.5	
issuerName	WS_2.5	
issuerRegionCode	WS_2.5	
panCheckAlgorithm	WS_2.5	
panLengthMax	WS_2.5	
panLengthMin	WS_2.5	

Tableau 99: Fields for the cardData element

6.5.2 EXAMPLE OF A CARD DATA CONSULTATION GETCARDDATA

6.5.2.1 Example of a getCardData request

```
{"cardNumber" : "4975000000000000","interfaceVersion" : "PMR_WS_2.5","keyVersion" :
"1","merchantId" : "011223344550000","seal" :
```

```
"97fccfbfde4570019236f31e3bb521be477bbc7bf7cf09c30f8fbc3a635a3011"}
```

6.5.2.2 Example of a getCardData response

```
{"paymentMeanInfoResponseCode":"00","cardDataList":[{"cardScheme":"CB","cardBrand":"CB","cardP
roductCode":"2","cardProductName":"CARTE NATIONALE DE RETRAIT ET DE
PAIEMENT","issuerCode":"10107","issuerCountryCode":"FRA","panLengthMin":16,"panLengthMax":16,"
panCheckAlgorithm":"L","cardProductProfile":"D"},{"cardScheme":"VISA","cardBrand":"VISA","card
ProductCode":"F","cardProductName":"VISA
CLASSIC","cardCorporateIndicator":"N","issuerCode":"10107","issuerCountryCode":"FRA","issuerRe
gionCode":"D","panLengthMin":16,"panLengthMax":16,"panCheckAlgorithm":"L"}],"seal":"c10bc66f57
4d22c4eb68250fe80c7d764a420a4fa22fde23f69e3a67bf60d131"}
```

6.6 DESCRIPTION OF THE TOKEN WEB SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/token
This web service allows you to use tokenize pan.

6.6.1 TOKENTOPAN SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/token/tokenToPan

This function makes it possible to transform a token to a pan.
Requests for validation operations include the following elements:

- default input elements
- tokenPanData element

6.6.1.1 Default inputs for the tokenToPan

Fields	Presence	As of version	Comments
merchantId	Mandatory	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed to TR_WS_2.0
userId	Optional	WS_2.0	
tokenPanDataList	Optional	WS_2.0	List of tokenPanData Container fields, see below
sealAlgorithm	Optional	WS_2.11	

Table 100: Fields for the tokenToPan request

6.6.1.2 tokenPanData inputs

Field	Presence	As of version	Comments
tokenPanId	Optional	WS_2.0	

tokenPan	Optional	WS_2.0	

Table 101: Fields for the tokenPanData element

6.6.1.3 tokenToPan outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
responseCode	WS_2.0	
reponsePanDataList	WS_2.0	List of responsePanData Container fields, see below

Table 102: Fields for the tokenToPan response

6.6.1.4 responsePanData outputs

Field	As of version	Comments
tokenResponseCode	WS_2.0	
tokenPanId	WS_2.0	
pan	WS_2.0	

Table 103: Fields for the responsePanData element

6.6.2 EXAMPLE OF A TOKENTOPAN INFORMATION REQUEST

6.6.2.1 Example of a tokenToPan request

```
{"interfaceVersion" : "TR_WS_2.0","keyVersion" : "1","merchantId" :
"011223344550000","panDataList" : [{"pan" : "490700h719850600","panId" : "1"}],"userId" :
"uid","seal" : "e350c058e2be7a3686682587dfac89cd192cda1ad2c2f532e10f9d59b9ca0f13"}
```

6.6.2.2 Example of a tokenToPan response

```
{"responseCode":"00","responsePanDataList":[{"tokenResponseCode":"00","tokenPanId":"1","pan":"
4907000000000600"}]}
```

6.6.3 PANTOTOKEN SERVICE

The URL is `https://office-server.sips-atos.com/rs-services/v2/token/panToToken`

Requests for validation operations include the following elements:

- default input elements
- panData element

6.6.3.1 panToToken default inputs

Fields	Presence	As of version	Comments
merchantId	Mandatory	WS_2.0	
interfaceVersion	Mandatory	WS_2.0	Value fixed at TR_WS_2.0
userId	Optional	WS_2.0	
panDataList	Optional	WS_2.0	List of panData Container fields, see below
sealAlgorithm	Optional	WS_2.11	

Table 104: Fields for the panToToken request

6.6.3.2 panData inputs

Field	Presence	As of version	Comments
panId	Optional	WS_2.0	
pan	Optional	WS_2.0	

Table 105: Fields for the panData element

6.6.3.3 panToToken outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
responseCode	WS_2.0	
responseTokenPanDataList	WS_2.0	List of responseTokenPanData Container fields, see below

Table 106: Fields for the panToToken response

6.6.3.4 responseTokenPanData outputs

Field	As of version	Comments
tokenResponseCode	WS_2.0	
tokenPan	WS_2.0	
panId	WS_2.0	

Table 107: Fields for the responseTokenPanData element

6.6.4 EXAMPLE OF A PANTOTOKEN INFORMATION REQUEST

6.6.4.1 Example of a panToToken request

```
{"interfaceVersion" : "TR_WS_2.0","keyVersion" : "1","merchantId" :
"011223344550000","panDataList" : [{"pan" : "4907000000000600","panId" : "1"}],"userId" :
"uid","seal" : "8f95bd9cc96c7723b530f0492c9fed50399b94f9b0c91c113b4f56a97fe2cb8f"}
```

6.6.4.2 Example of a panToToken response

```
{"responseCode":"00","responseTokenPanDataList":[{"tokenResponseCode":"00","panId":"1","tokenP
an":"490700h719850600"}],"seal":"a16689b166fa9a508341fad15563633bb83367d009d3f2507c95bc71cd9f1
ad5"}
```

6.7 Description of the Fraud web service

The URL is https://office-server.sips-atos.com/rs-services/v2/fraud
This web service allows you to check on existing transaction.

6.7.1 GETVELOCITYDATA SERVICE

The URL is https://office-server.sips-atos.com/rs-services/v2/fraud/getVelocityData

This function makes it possible to check the activity of a given data field over a given period.
Requests for validation operations include the following elements:

- default input elements

6.7.1.1 Default inputs for the getVelocityData

Fields	Presence	As of version	Comments
interfaceVersion	Mandatory	WS_2.0	Value fixed to FR_WS_2.9
keyVersion	Mandatory	WS_2.0	
merchantId	Mandatory	WS_2.0	
velocityElementType	Mandatory	WS_2.0	
velocityElementValue	Mandatory	WS_2.0	
velocityPeriod	Optional	WS_2.0	
sealAlgorithm	Optional	WS_2.11	

Table 108: Fields for the getVelocityData request

6.7.1.2 getVelocityData outputs

Responses include the following elements:

- Default outputs

Fields	As of version	Comments
responseCode	WS_2.0	
velocityTotalAmount	WS_2.0	
velocityNbTransaction	WS_2.0	
currencyCode	WS_2.0	
velocityProfileMaxTotalAmount	WS_2.0	
velocityProfileMaxNbTrans	WS_2.0	
velocityProfileMaxTransAmount	WS_2.0	
velocityProfileName	WS_2.0	
velocityProfilePeriod	WS_2.0	
velocityProfileDateTime	WS_2.0	

Table 109: Fields for the getVelocityData response

6.7.2 EXAMPLE OF A GETVELOCITYDATA INFORMATION REQUEST

6.7.2.1 Example of a getVelocityData request

```
{"interfaceVersion" : "FR_WS_2.9","keyVersion" : "1","merchantId" :
"011223344550000","velocityElementType" : "customerId","velocityElementValue" :
"cust010","velocityPeriod" : "50","seal" :
"fc24b9e6bd7a433d258a1df1a8af2698b917d1632aaa1bc12c5c8f45adbeba11"}
```

6.7.2.2 Example of a getVelocityData response

```
{"responseCode":"00","velocityTotalAmount":0,"velocityNbTransaction":0,"velocityProfilMaxTotal
Amount":1000,"velocityProfilMaxNbTrans":10,"velocityProfilMaxTransAmount":100,"currencyCode":"
978","velocityProfileName":"all_controls","velocityProfilPeriod":10,"velocityProfileDateTime":
"2014-11-
19T14:21:32+01:00","seal":"5a65b70f786047ef13f80a55a6642211daa52df74039c605e70ac7d752f52731"}
```

END OF DOCUMENT