SOAP Web service API Payment order

Implementation Guide

Document version 1.7

Contents

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.7	Natixis Payment Solutions	1/4/2018	• Updating **Creating a payment order by e-mail**. Codes samples fixed.
1.6	Natixis Payment Solutions	11/7/2017	Several fixes added int the document.
1.5	Natixis Payment Solutions	10/11/2017	Addition of usecases of the expandedData field. Several precisions added int the document.
1.4	Natixis Payment Solutions	9/22/2017	**paymentOfferInfo** and **paymentOfferEntity** : Addition of the **template** field.
1.3	Natixis Payment Solutions	2/3/2017	**paymentOfferEntity** : name of the **recipients** field changed with **recipient**.
1.2	Natixis Payment Solutions	6/21/2016	Update (DITA format) Addition of the **expandedData** field.
1.1	Natixis Payment Solutions	3/6/2013	Updated documentation.
1.0	Natixis Payment Solutions	6/24/2011	Initial version.

2. GETTING IN TOUCH WITH TECHNICAL SUPPORT

For technical inquiries or support, you can reach us from Monday to Friday, between 9am and 6pm

by phone at:	0 811 363 364 Service 0,06 € / min + prix appel for the clients of Banque Populaire
by phone at:	0 810 004 724 Service 0,06 € / min + prix appel for the clients of Caisse d'Epargne
by e-mail :	*supportvad@lyra-network.com*

To facilitate the processing of your demands, you will be asked to communicate your shop ID (an 8-digit number) .

This information is available in the "registration of your shop" e-mail or in the Merchant Back Office (**Settings** > **Shop** > **Configuration**).

3. INTRODUCING WEB SERVICES PAYMENT ORDER BY E-MAIL

Web services are developed according to the version 1.2 of the SOAP protocol (Simple Object Access Protocol) and are described by the following wsdl file :

https://paiement.systempay.fr/vads-ws/paymentoffer-v2?wsdl

If you use a proxy to connect to the Internet via an application server, contact your system administrator to make sure whether it is necessary to configure the access to this URL.

In order to secure the exchange, it is encrypted according to the HTTPS protocol. Furthermore, a signature mechanism has been set up to validate and authenticate the exchange of data.

Description of the service

This service allows to automate the creation / modification of a payment order to one or more recipients.

For it, two methods are available :

- *create* for the creation of a payment order by e-mail.
- *update* for the modification of a payment order by e-mail.

The response returned by the server contains the list of created / modified orders.

Each order is associated with a numeric ID, an e-mail address and a payment URL.

Sending to a list of recipients :

It is possible to define a list of recipients in the request.

In this case, one payment order is created for each recipient. There are therefore as many payment order as recipients.

Thus, after the payment, each transaction match an unique recipient.

In order to identify all created orders, the **reference** field is incremented by 1 in 3 digits.

These characters are concatenated to the reference with a hyphen. Example : [ref]-001, [ref]-002, [ref]-003.

Example of a payment order created with the **reference** field filled with"Recovery" :

If the order is sent to 3 recipients, then 3 orders will be created with following references :

- Recovery-001
- Recovery-002
- Recovery-003

Use of an E-mail marketing tool :

If the merchant wish to use his own mailing tool, he must disable the automatic sending in the payment order creation / modification requests.

By analyzing the response, he will find all required datas to fill the e-mail : payment URL, order reference and recipient e-mail address.

4. IDENTIFY THE TYPES USED FOR QUERIES

4.1. paymentOfferInfo

The **paymentOfferInfo** type allows to describe the parameters for a **creation** of a payment order by e-mail.

Field name	Type	Description	Required
shopId	Long n8	Shop ID	✓
reference	string an24	Transaction ID or order reference	
ctxMode	string	Defines the mode of interaction with the payment gateway. • TEST • PRODUCTION	✓
amount	long	The amount of the transaction presented in the smallest unit of the currency (ex: cent for Eur).	✓
currency	int	Currency code of the transaction (ISO 4217 standard). E.g.: 978 for euro (EUR)	✓
locale	string	Language code used to send notifications (e-mail or SMS payment confirmation) List of available languages **Language** / **ISO 639-1 standard** German / de English / en Chinese / zh Spanish / es French / fr Italian / it Japanese / ja Dutch / nl Polish / pl Portuguese / pt Russian / ru Swedish / sv	✓
message*	string an2000	Body of the e-mail message sent.	✓
recipients	Array [1-100]	List of e-mail addresses. An order is created by recipient (min 1, max 100).	✓
subject*	string an255	Subject of the e-mail message sent.	✓
validationMode	int	Payment validation mode : • 0 = Automatic (by default) • 1 = Manual.	✓
validity	dateTime	Validity date of the order. Can not be less than the current date and may not exceed 90 days. Date in ISO 8601 format defined by W3C. *Example : 2016-07-16T19:20:00Z.*	✓
sendMail	boolean	Sends the e-mail to the recipient if true.	✓
expandedData	string	Allows to add dynamically any of the form field. *Examples:*	

Field name	Type	Description	Required

		Create a payment order n times and in English
		`vads_payment_config=MULTI:first=1000;count=3;period=30&vads_language=en`
		Propose the registration of the card during payment
		`vads_page_action=ASK_REGISTER_PAY`
		Make a one-click payment
		`vads_identifier=968533214746354778521330 1`	

device	string	Possible value = « MAIL »	✔
template	string	Name of the e-mail template used for the payment order.	

Table 1: paymentOfferInfo

* The fields ***message*** and ***subject*** become optional when ***template*** is filled.

Initialization example with JAVA

```
private PaymentOfferInfo initInfo()
{
 PaymentOfferInfo info = new paymentOfferInfo();
 info.setShopId(PDV_ID);
 info.setAmount(10000);
 info.setCurrency(978);
 info.setCtxMode("PRODUCTION");
 info.getRecipients().add(MAIL1);
 info.getRecipients().add(MAIL2);
 info.setLocale("fr");
 info.setMessage(BODY);
 info.setReference("REF-ORDRE");
 info.setDevice(PaymentDevice.MAIL.toString());
 info.setSubject(SUBJECT);
 info.setValidationMode(0);
 info.setValidity(UtilWsTests.getNewDate(90));
 info.setSendMail(true);
 info.setExpandedData("vads_payment_config=MULTI:first=1000;count=3;period=30&vads_language=en");
 return info;
}
```


4.2. paymentOfferEntity

The **paymentOfferEntity** type allows to describe the parameters for the **modification** of a payment order by e-mail.

the difference with *paymentOfferInfo* is that it contains the order ID and the email address of the order recipient.

Field name	Type	Description	Required
shopId	Long n8	Shop ID	✓
offerId	Long an32	Order reference	✓
reference	string an24	Transaction ID or order reference	
ctxMode	string	Defines the mode of interaction with the payment gateway. • TEST • PRODUCTION	✓
paymentURL		URL of the payment order	
amount	long	The amount of the transaction presented in the smallest unit of the currency (ex: cent for Eur).	✓
currency	int	Currency code of the transaction (ISO 4217 standard). E.g.: 978 for euro (EUR)	✓
locale	string	Language code used to send notifications (e-mail or SMS payment confirmation) List of available languages **Language** / **ISO 639-1 standard** German / de English / en Chinese / zh Spanish / es French / fr Italian / it Japanese / ja Dutch / nl Polish / pl Portuguese / pt Russian / ru Swedish / sv	✓
message*	string an2000	Body of the e-mail message sent.	✓
recipient	string	E-mail of the recipient.	✓
subject*	string an255	Subject of the e-mail message sent.	✓
validationMode	int	Payment validation mode : • 0 = Automatic (by default) • 1 = Manual.	✓
validity	date	Validity date of the order. Can not be less than the current date and may not exceed 90 days. Date in ISO 8601 format defined by W3C. *Example : 2016-07-16T19:20:00Z.*	✓
sendMail	boolean	Sends the e-mail to the recipient if true.	✓
expandedData		Allows to add dynamically any of the form field. *Examples:*	

Field name	Type	Description	Required

		Create a payment order n times and in English
		`vads_payment_config=MULTI:first=1000;count=3;period=30&vads_language=en`
		Propose the registration of the card during payment
		`vads_page_action=ASK_REGISTER_PAY`
		Make a one-click payment
		`vads_identifier=9685332147463547785213301`	

device	string	Possible value = « MAIL »	✔
template	string	Name of the e-mail template used for the payment order.	

Table 2: paymentOfferEntity

* The fields ***message*** and ***subject*** become optional when ***template*** is filled.

5. IDENTIFY THE TYPES USED FOR RESPONSE

5.1. paymentOfferResponse

The **paymentOfferResponse** type is the response object returned by the server during after a **creation** or **modification** request.

If the operation is successful, the object contains a list of orders IDs created or modified if not it contains an error code.

Field name	Type	Description
offerEntities	list[1-100]	List of created orders expressed as a list of paymentOfferEntity objects.
reponseCode	String	Response code sent by the server.
returnMessage	String	English description of the *reponseCode* field.
extendedCode	String	Returned in TEST mode in case of wrong signature calculation. Describes the chain without the certificate used to compute the signature before theSHA1 hash.

Table 3: paymentOfferResponse

5.2. Description of response codes

responseCode	Description
OK	Action successfully completed.
NOT_ALLOWED	Action not authorized.
NOT_AUTHORIZED	Changing the order specified is prohibited. *Note*: *The modification of an order is possible in two cases:* • *No Payment associated with this order* • *One or more declined payments associated with this order.*
OFFER_NOT_FOUND	The order ID does not exist.
BAD_SIGNATURE	Bad signature.
RECIPIENTS	The recipients list may not be null or empty and must not exceed the number of 100 e-mail addresses.
RECIPIENT	The specified e-mail address is invalid.
SUBJECT	The subject of the message can not be null.
MESSAGE	The body of the e-mail can not be null.
CTX_MODE	The connection mode can be only "TEST" or "PRODUCTION".
DEVICE	Can be only populated with "MAIL".
VALIDITY_DATE	The *validity* parameter is invalid.
AMOUNT	The *amount* parameter is invalid.
VALIDATION_MODE	The *validationMode* parameter is invalid.
SYSTEM_FAILURE	Server error
TEMPLATE_NOT_FOUND	The e-mail template does not exist.

Table 4: responseCode

6. IDENTIFY HOW TO MANAGE A PAYMENT ORDER

6.1. Creating a payment order by e-mail

Important :

- *If **sendMAIL**= **true**, an e-mail is received by each buyer with a single payment URL.*
- *If **sendMAIL**= **false**, the buyer will not receive e-mail. The merchant has the task to chose the way to pass the payment URL on to the buyer.*

The **create** function allows to create a payment order. It takes as input the following parameters:

Field name	Type	Description	Required
info	paymentOfferInfo	Parameters for the creation of a payment order (see **paymentOfferInfo**)	✔
wsSignature	string	Signature	✔

The signature computation is done using the parameters in the following order:

shopId, *reference*, *ctxMode*, *amount*, *currency*, *locale*, *message*, *recipients*, *subject*, *validationMode*, *validity*, *sendMail*, *expandedData*

Notes :

In the signature computation:

- The **dateTime** fields must be formated in YYYYMMDD.

  Example : 2016-06-22T23:00:00+00:00 becomes 20160622

- The string field **recipients** should be in brackets [...]. The e-mail addresses must be separated by a comma and a space.

  Examples:

  [email]

  [email1, email2, email3]

The **create** function returns a **paymentOfferResponse** response type.

Example of XML generated file

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=""
 xmlns:ns1="">
 <SOAP-ENV:Body>
  <ns1:create>
   <info>
    <shopId>12345678</shopId>
    <ctxMode>TEST</ctxMode>
    <subject>Your payment order</subject>
    <message>Hello, please find below a payment link</message>
    <device>MAIL</device>
    <reference>CMD123</reference>
    <validity>2016-06-22T23:00:00+00:00</validity>
    <amount>10000</amount>
    <validationMode>0</validationMode>
    <currency>978</currency>
    <locale>fr</locale>
    <sendMail>true</sendMail>
    <recipients>test@test.fr</recipients>
```



```
    <recipients>test1@test.fr</recipients>
    <expandedData>vads_payment_config=MULTI:first=1000;count=3;period=30&vads_language=en</
expandedData>
   </info>
   <signature>1ee15221a1b9c062e01ba4a790e46581418f72fb</signature>
  </ns1:create>
 </SOAP-ENV:Body>
 </SOAP-ENV:Envelope>
```

Corresponding signature computation:

12345678+CMD123+TEST+10000+978+fr+Hello, please find below a payment link+[test@test.fr, test1@test.fr]+Your payment order+0+20160622+1+vads_payment_config=MULTI:first=1000;count=3;period=30;vads_language=en+certificat

Example of XML generated file with use of e-mail template

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap=""
 xmlns:v2="">
  <soap:Header/>
 <soap:Body>
  <v2:create>
  <info>
   <shopId>12345678</shopId>
   <ctxMode>TEST</ctxMode>
   <reference>CMD123</reference>
   <template>templateTest</template>
   <device>MAIL</device>
   <validity>2017-12-10T19:20:00Z</validity>
   <amount>10000</amount>
   <validationMode>0</validationMode>
   <currency>978</currency>
    <locale>fr</locale>
   <sendMail>true</sendMail>
   <recipients>test@test.fr</recipients>
   <expandedData>vads_payment_config=MULTI:first=1000;count=3;period=30</expandedData>
   </info>
    <signature>719595eecd97ec87de35fb6973ab425b6d45ab61</signature>
  </v2:create>
    </soap:Body>
</soap:Envelope>
```

Corresponding signature computation:

12345678+CMD123+TEST+10000+978+fr
+[test@test.fr]+0+20171210+1+vads_payment_config=MULTI:first=1000;count=3;period=30+certificate

6.2. Modify a payment order by e-mail

The **update** function allows to create a payment order. It takes as input the following parameters:

Field name	Type	Description	Required
entity	paymentOfferEntity	Parameters for the modification of a payment order (see **paymentOfferEntity**)	✔
wsSignature	string	Signature	✔

The signature computation is done using the parameters in the following order:

shopId, *offerId*, *reference*, *ctxMode*, *amount*, *currency*, *locale*, *message*, *recipients*, *subject*, *validationMode*, *validity*, *sendMail*, *expandedData*

Notes :

In the signature computation:

- The **dateTime** fields must be formated in YYYYMMDD.

  Example : 2016-06-22T23:00:00+00:00 becomes 20160622

- The string field **recipients** should be in brackets [...]. The e-mail addresses must be separated by a comma and a space.

  Examples:

  [email]

  [email1, email2, email3]

The **update** function returns a **paymentOfferResponse** response type.

7. UNDERSTANDING THE COMPUTATION METHOD OF THE SIGNATURE

A key (or "certificate") is required to communicate with the payment gateway.

Two certificates are available in your Back Office from the menu **Settings** > **Shop** > **Certificates** tab :

- one for TEST mode,
- one for PRODUCTION mode.

The key value is used to compute the alphanumeric signature.

To compute the signature:

- The **Numeric** type fields can't have a 0 to the left of the most significant digit.
- The **bool** type fields take the values 1 (true) or 0 (false)
- The unfilled **string** type fields will be empty

To simplify the signature calculation, all dateTime fields and the template field are ignored.

1. Respect the order of the fields :
   - Creation of a payment order :

   shopId, *reference*, *ctxMode*, *amount*, *currency*, *locale*,*message*, *recipients*, *subject*, *validationMode*, *validity*, *sendMail*, *expandedData*

   - Modification of a payment order :

   shopId, *offerId*, *reference*, *ctxMode*, *amount*, *currency*, *locale*,*message*, *recipients*, *subject*, *validationMode*, *validity*, *sendMail*, *expandedData*
2. Make sure that all the fields are encoded in UTF-8.
3. Concatenate the values of these fields separating them with the "**+**" character.
4. Concatenate the result with the test or production key separating them with a "**+**".
5. Apply the SHA-1 algorithm to obtain the signature value.

In TEST mode, n case of bad signature computation, the **errorCode** returned is "2" and the string used for signature on the server is then returned in the **errorDetail** field.

8. APPENDIX

8.1. Fields sent during Instant Payment Notification

Several types of notifications are available in the Back Office. They allow the configuration of the URL of the page to contact and the management of the events (payment abandoned by the buyer, payment canceled by the merchant, payment validated by the merchant, etc.) that will trigger a call to the merchant website.

This call contains all response parameters related to payment. The parameters are described in the **Payment Gateway implementation guide** available on our documentation website.

Among the response parameters, there are :

Variable	Description	Value
vads_url_check_src	Payment resulting from a payment order by phone.	**VOICE_ORDER**
vads_token_id	Set by the public ID of the payment order.	

8.2. User variables

Variables	Description	Required
%reference%	Matches with the **Reference** field entered in the **Order details** group.	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*
%amount%	Matches with the **Amount** field entered in the **Order details** group.	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*
%start_date%	Matches with the payment order creation date. ***Note:*** *When editing a payment order, this variable is set to the creation date registered at the time of the creation. This value cannot be edited.*	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*
%end_date%	Matches with the **Expiry date** field entered in the **Order details** group.	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*
%shop_name%	Matches with the name of the sender. This variable is by default used in the message **Message sent by %shop_name%**. The displayed name is the shop name defined in the **Configuration** tab (Shop settings). The text **Message sent by** can be customized.	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*
%url%	Link to perform the payment. The link specifies: • the amount • the expiry date of the payment order By clicking on the link, the payment page is displayed.	Yes ***Note:*** If the **%url%** variable is not entered in the body of the e-mail, this link is automatically added at the end of the text.
%shop_contact%	E-mail of the sender.	No ***Note:***

Variables	Description	Required
	This variable is by default used in the message **In case of problems, or if this message is not correctly displayed, please contact %shop_contact%**. The e-mail address displayed is the "e-mail manager " registered during the shop registration. If the e-mail manager is not entered, the e-mail of the sender will be displayed. Please contact the Systempay costumer service if you would like to modify it. The message **In case of problems, or if this message is not correctly displayed, please contact** is customizable.	*The value of the variable is automatically resolved in the object and the text of the e-mail.*
%shop_url%	Corresponds to the merchant website URL. This variable is by default used in the message **Access to online shop: %shop_url%**". The displayed URL is the link to return to the merchant website. The message **Access to online shop:** is customizable.	No ***Note:*** *The value of the variable is automatically resolved in the object and the text of the e-mail.*

Table 5: Variables

