Paylane Direct System

Webservice based payment management system

Created by: PayLane IT Crew / 2005-05-12

Last modification: 2012-10-05

Saved by: Jan Makulec

Table of contents

I. System overview

PayLane Direct System is accessible using a set of webservices that are available for PayLane Merchants to perform various actions directly on the PayLane Payment Processing System.

The webservices can be accessed over secure web connections only (HTTPS) and it is not possible to access them using insecure web connections (HTTP).

Additionally, an authentication mechanism is in place. Webservices are available only to registered servers (IP numbers) after supplying a unique login/password pair that is assigned to each merchant account.

I.I. Feature comparison

PayLane Direct System is an advanced integration method. It provides the most features and is the most configuration-rich at a cost of investing a little more effort into technical integration.

If you need a simple solution, make sure to read “PayLane Secure Form” document, downloadable from PayLane Merchant Panel (Support > Documentation).

The table below presents full feature comparison between PayLane Direct System and PayLane Secure Form.

Feature	**PayLane Direct System**	**PayLane Secure Form**
Access method	SOAP protocol (web service)	simple POST form
Gathering sensitive customer data	merchant side	PayLane side
SSL certificate required	yes	no
3-D Secure Authentication	yes (merchant & PayLane side)	yes (PayLane side)
Card transactions	yes	yes
Direct Debit transactions	yes	yes
Authentication+capture mode	yes	yes (capture only from Merchant Panel)

Refunds	yes	yes (only from Merchant Panel)
Chargebacks handling	yes	yes
Transaction Notifications	yes (optional)	yes (optional)
Email Notifications	yes (configured in Merchant Panel)	yes (configured in Merchant Panel)
Fraud Protection	yes (configurable)	yes
Blacklists	yes	yes
PayPal	yes	yes

No matter what integration method you choose, all universal features such as advanced fraud prevention, multi-currency support or access to Merchant Panel will be available to you.

2. Basic information

2.1. Registering servers in PayLane Direct System

Before making any queries to the webservices you need to register your servers with PayLane Direct System. You need to supply PayLane with the list of hosts (IP numbers) that will have access to the webservices. In order to add the IP numbers, follow this path in the merchant panel (http://merchant.paylane.com): Account => Merchant account settings => Direct => Edit.

Note that hosts are registered to merchant accounts – a separate registration is required for each merchant account you have with PayLane.

2.2. SOAP based webservices

All webservices are accessible using the SOAP interface. You can access webservices from virtually any programming language that has SOAP support.

All webservices are described using WSDL – your SOAP client should be able to read and understand WSDL files in order to make function calls within the web service.

If you cannot use WSDL files you can still access webservices – in this case you need to read WSDL files and create all the functions yourself.

2.3. Encryption

All the connections to webservices are encrypted using SSL. Therefore it is only possible to access webservices over the secure HTTPS connections.

Note: WSDL files are also available only via HTTPS connection. It is not possible to access WSDL files over insecure connection.

2.4. Authentication

Access to webservices is protected by username/password pair using the HTTP AUTH method. The username/password will be supplied by PayLane. If you have more than one account then you will receive separate login/password pairs for each of your accounts.

Please also be aware that you have to supply PayLane with all your IP addresses that will be used to access PayLane webservices. Please send the list of IP addresses to PayLane Technical Support (support@paylane.com) or follow this path in the merchant panel to add the IP addresses yourself: Account => Merchant account settings => Direct => Edit. Remember to always update PayLane when your IP addresses change.

2.5. Test mode

When integrating your application with PayLane Payment Processing System you will perform various tests. To enable you execute them, you will be provided with a test account that you can use both in the integration phase and after, when you test changes in your applications.

Remember that the test account will be enabled after the integration phase to let you easily test changes in your applications.

2.5.1. Test mode card numbers

While using test account, you may use any card number you want that passes "Luhn test" to test credit card transactions. Such numbers may be easily generated for various card vendors (Visa, MasterCard etc). PayLane test mode will simulate successful charge for such card numbers.

Some example card numbers that can be used for testing purposes are provided in Appendix B (that includes special test card numbers that can be used to simulate various payment process errors and other situations).

2.5.2. Test mode amount triggers

In addition to test card numbers, you may also use amount triggers. For example: if you enter "501.00" as amount, PayLane will simulate error 501 etc.

Amount triggers will work for every card number (they are independent from test card numbers) and can be used to test every method (`multiSale`, `refund` etc).

2.6. Production mode

After successfully completing the integration you will be sent the details of a separate production account to perform real sales.

2.7. Address Verification System (AVS)

Every credit card can be passed through Address Verification System (AVS). AVS contacts card issuer (bank) and checks if address (street) and ZIP code entered by cardholder match the card number.

AVS is supported only in United States, Canada (VISA, MC) and Great Britain (VISA, Maestro).

You can choose how restrictive AVS will be (what qualifies a transaction to be rejected because of AVS mismatch).

AVS levels available at PayLane (applies to all cards but Maestro UK)

Level	Rejected AVS results*	Description
0	-	AVS is disabled.
1	N	If both address and ZIP mismatch, transaction is rejected.
2	ABN	If ZIP mismatches, transaction is rejected.
3	IPWZABN	If ZIP or address mismatches, transaction is rejected.
4	RIPWZABN**	Most restrictive. Both address and ZIP match are required. Cards that cannot be verified using AVS are rejected.

* See Appendix A for list of all AVS results.

** Level 4 will reject ALL cards issued outside US/Canada/GB. At levels 0-3, if card issuer does not support AVS, AVS result is ignored.

Default AVS level is configured globally for merchant account by PayLane Support. Merchant account configuration also allows (or disallows) overriding default AVS level with every transaction.

Maestro UK / Solo cards are treated differently than other cards. AVS is mandatory for all Maestro UK transactions. You also can not change default rejection policy.

For Maestro UK, AVS result is a 2-digit code. First digit is a result of post code check, the second is a result of address check (see Appendix A for all possible combinations):

Result	Description
1	Not checked.
2	Match.
4	No match.
5	Partial match.

Following AVS results for Maestro UK / Solo will cause rejecting the transaction:

AVS result	Description
11	Post code and address were not checked.
14	Post code was not checked, address does not match.
41	Post code does not match, address was not checked.
44	Post code and address do not match.

2.8. 3-D Secure

PayLane provides full native support for 3-D Secure Authentication of card transactions.

3-D Secure is an emerging industry standard, supported by Visa (Verified by Visa) and MasterCard (MasterCard SecureCode). In 3-D Secure transactions cardholder is required to provide additional authentication on cardholder's bank website.

3-D Secure engages three parties: merchant, card issuer (bank) and card schemes (Visa/MasterCard) to provide additional security to transactions and prevent fraud and chargebacks.

PayLane strongly encourages all merchants to implement and use 3-D Secure mechanisms.

Detailed guide to performing 3-D Secure transactions with PayLane Payment Processing System is available as a separate document.

3. Description of webservices

3.1. WSDL location

There is one WSDL file that includes a complete definition of the webservice:

https://direct.paylane.com/wsdl/production/Direct.wsdl

3.2. SOAP Functions

3.2.1. multiSale

This method is used to perform or initiate a sale using various payment methods.

If sale authorization is performed, remember that captureSale method needs to be used later to complete (capture) the sale.

Currently there are three payment methods available:

- card charge
- direct debit
- redirect transaction

If you are not sure which method to choose when performing sales, please contact PayLane Technical Support (support@paylane.com) to get more information.

Field	Format	Description
`multi_sale_params`	structure (xsd:complexType)	List of parameters that will be used to perform a transaction. The exact format of this field can differ depending on the SOAP client implementation of the programming language and is usually either an object with properties or an array with named values.

`multi_sale_params` structure explained:

Attribute	Required	Format	Description
`payment_method`	Y	structure (xsd:complexType)	Structure with payment method details
`capture_later`	N	boolean	If true, only sale authorization (funds reservation) will be performed. Use `captureSale` method to complete the sale. Default = `false`. *
`fraud_check_on`	N	boolean	Overrides default setting for merchant account. If true, sale data

			will be checked for fraud, if false, data will not be checked. **
`avs_check_level`	N	integer (0..4)	Overrides default AVS level. Levels from 0 (disabled) to 4 (all required) can be set. ***
`customer`	Y	structure (xsd:complexType)	Structure with customer data. Explained in details later in this document.
`amount`	Y	decimal(12,2)	Total amount of the sale. Decimal point is a dot. Example: 134.25
`currency_code`	Y	string(3)	Currency of the sale. Format of the currency as in the ISO 4217 standard. All uppercase.
`processing_date`	N	string(10)	
`Product`	Y	structure (xsd:complexType)	Structure containing product/sale additional information

* Authorization and capture model is only possible for card payments. Specifying `capture_later = true` for direct debit sales will result in error 407.

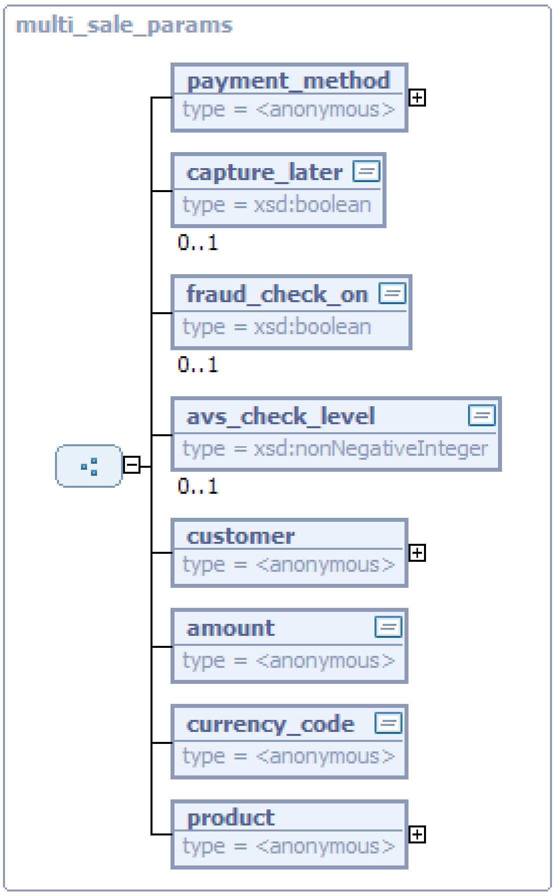
** Fraud check can only be performed for card payments. Specifying `fraud_check_on = true` for direct debit sales will result in error 408.

Overriding default fraud check status is only possible if the merchant account is configured to allow it. Otherwise, error 409 is returned.

*** AVS (Address Verification System) is only available for card payments. Specifying `avs_check_level` different than 0 (disabled) will result in error 408.

Overriding default AVS level is only possible if the merchant account is configured to allow it. Otherwise, error 409 is returned. Overriding does not apply to Maestro UK cards.

multi_sale_params structure overview:



payment_method structure explained:

Attribute	Required	Format	Description
`card_data`	Y*	structure (xsd:complexType)	Card details. Provide this field if you want to select Card Charge as a payment method.
`account_data`	Y*	structure (xsd:complexType)	Bank account details. Provide this field if you want to select Direct Debit as a payment method.

* `card_data` and `account_data` fields are interchangeable. You must provide either `card_data` field (to perform a card charge) or `account_data` field (to perform a Direct Debit). Providing both fields at the same time (or neither) is considered an error.

card_data structure explained:

Attribute	Required	Format	Description
`card_number`	Y	13-19 digits	The full number of the card without any whitespaces. Only numbers are allowed.
`card_code`	Y	3-4 digits	Depending on the type of the card this will be either CVV2 (Visa), CVC2 (MasterCard) or CID (American Express).
`issue_number`	N*	1-3 digits	Issue number is present on some cards (e.g. some Maestro UK cards).
`expiration_month`	Y	01..12	Expiration month. 01 to 12.
`expiration_year`	Y	2008..2018	Expiration year as on the card. 2008 to 2018.
`name_on_card`	Y	UTF-8 encoded string, 2-50 characters	Cardholder name as written on the card.
`secure3d`	N**	structure (xsd:complextype)	3-D Secure Authentication data. Required only if 3-D Secure Authentication was performed before the sale.

* `issue_number` may be required for some Maestro UK cards only. When performing Maestro UK transactions it is advisable to ask cardholders for the issue number.

** `secure3d` structure is required whenever transaction 3-D Secure Authentication was initiated using `check3DSecureCardEnrollment` method (even if card was found as not participating in 3-D Secure).

secure3d structure explained:

Attribute	Required	Format	Description
`id_secure3d_auth`	Y	unsigned long	ID of 3-D Secure Authentication, as provided by check3DSecureCardEnrollment method. Detailed description of

			performing 3-D Secure enabled sales is provided in a separate document by PayLane.

account_data structure explained:

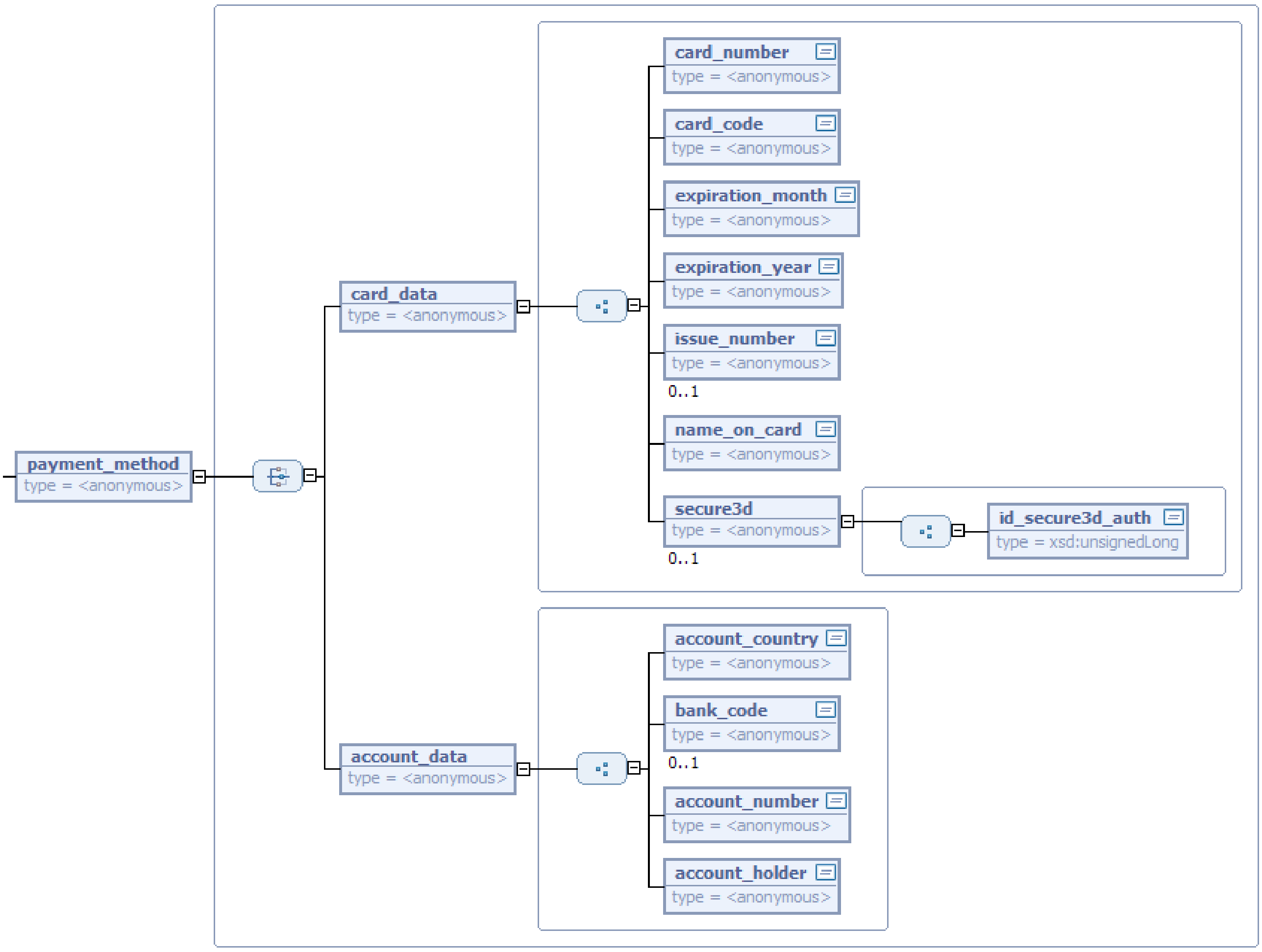
Attribute	Required	Format	Description
`account_country`	Y	2 chars	Bank account country code (as in ISO 3166).
`bank_code`	N*	UTF-8 encoded string, 5 or 8 characters	Bank Identifier Code (SWIFT/BIC/BLZ) Empty for the Dutch banks 5 chars for the Austrian banks 8 chars for the German banks
`account_number`	Y	UTF-8 encoded string, 1-11 characters	Account number.
`account_holder`	Y	UTF-8 encoded string, 2-30 characters	Account holder name.

* `bank_code` is required for Austrian and German accounts.

redirect_data structure overview:

Attribute	Required	Format	Description
`payment_type`	Y	2 chars	2 characters bank code: • 'IN' – Inteligo • 'IP' - iPKO • 'MT' – mTransfer • 'MU' – multiTransfer • 'DB' - Deutsche Bank • 'AB' - Alior Bank • 'AS' - Alior Sync • 'MI' - Millenium • 'CA' - Credit Agricole • 'OH' - Other
`back_url`	Y	UTF-8 encoded string	Redirect URL.

payment_method structure overview:



customer structure explained:

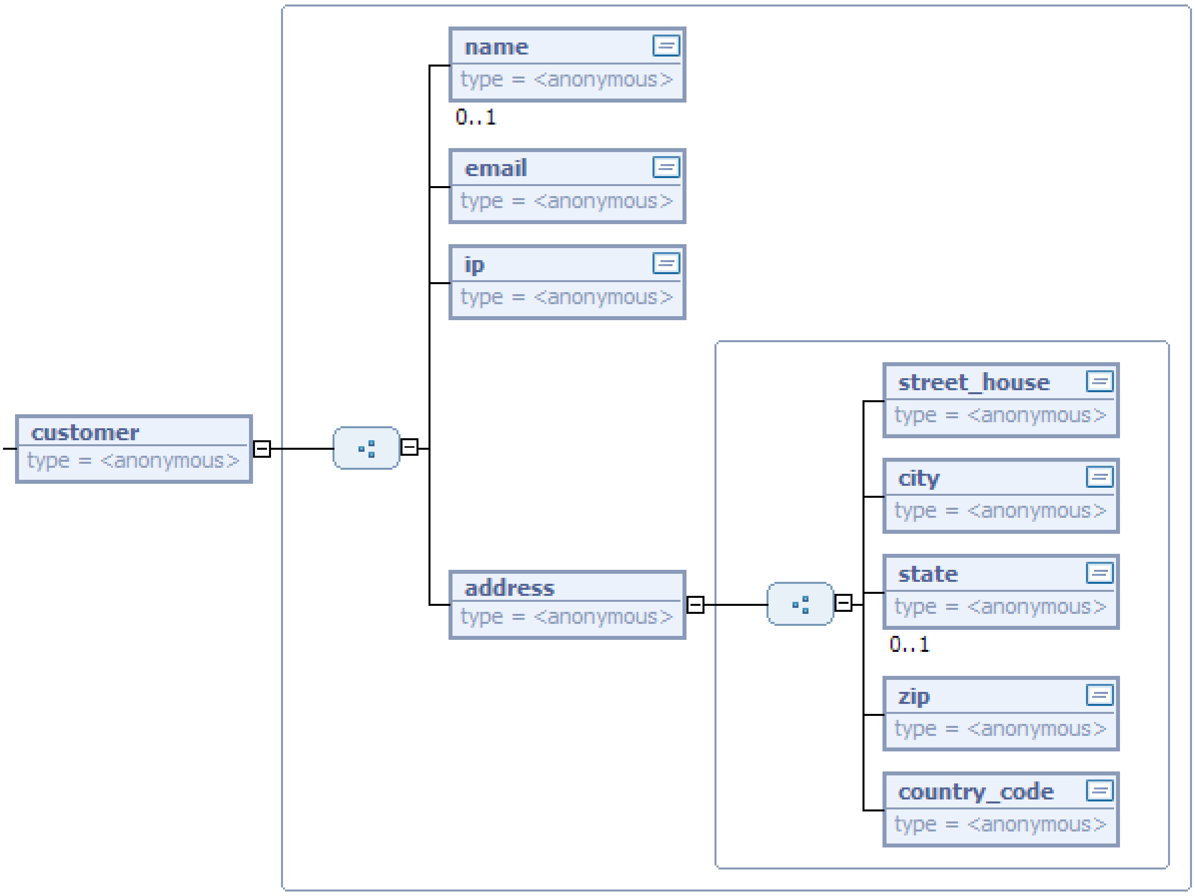
Attribute	Required	Format	Description
`name`	N	UTF-8 encoded string, 2-50 characters	Customer name or company name. Usually the same as in the `name_on_card` or `account_holder` field but can be different.
`email`	Y	string, 6-80 characters	Valid e-mail address.

`ip`	Y	string, max 15 chars, only digits and dots	The IP address of the client who performs a transaction. Should be always a routable IP number, not a private IP number. If you supply a private IP number it is likely that the sale will be declined by the anti fraud system.
`address`	Y	structure (xsd:complexType)	Structure containing address information of the client who performs a transaction.

address structure explained:

Attribute	**Required**	**Format**	**Description**
`street_house`	Y	UTF-8 encoded string, 2-46 characters	Street name, house number along with the apartment number if applicable.
`city`	Y	UTF-8 encoded string, 2-40 characters	City.
`state`	N	UTF-8 encoded string, 2-40 characters	State or province if applicable. It is not required but is recommended to maintain full address information.
`zip`	Y	UTF-8 encoded string, 1-9 characters	Zip or postal code. If not applicable please enter n/a.
`country_code`	Y	2 chars, according to the ISO 3166 standard	Country code (ISO 3166 standard)

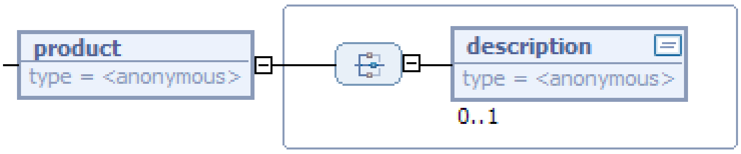
customer structure overview:



product structure explained:

Attribute	Required	Format	Description
`description`	N	UTF-8 encoded string, 2-200 characters	A short description of the sale that should briefly tell what the sale is for. It will be shown along on the sale details page in the Merchant Panel. It is strongly recommended to put unique identification information here.

product structure overview:



Response object attributes:

Field	Format	Description
`OK`	structure (xsd:complexType)	Present only if sale succeeded. Elements OK and ERROR are interchangeable.
`ERROR`	structure (xsd:complexType)	Present only if sale failed. The reason will be expressed using the error number and a description. All error codes can be found in a separate table below.
`DATA`	structure (xsd:complexType)	Optional additional sale information such as fraud score. Also present when sale fails.

Note that all Direct Debit sales are not authorized in real-time, which means that there is no real time capture of the funds. In order to check if the Direct Debit sale funds capture was successful you should query the PayLane Direct System for a current transaction status using the `checkSales` method.

Card sales perform fund capture immediately, unless you specify parameter `capture_later = true`. In this case `captureSale` method can be used to complete such sales.

OK object attributes:

Attribute	Format	Description
`id_sale*`	unsigned long	Identification number of the sale. This number should be always saved in your system to perform other operations on a sale (for example refunds). Sale identification number is unique for each sale.
`id_sale_authorization*`	unsigned long	Identification number of the sale authorization. This number is necessary to perform capture and/or close the authorization and should be saved.

`redirect_url**`	UTF-8 encoded string	The URL address where the customer is redirected.

* `id_sale` and `id_sale_authorization` are interchangeable. If you perform regular sale (parameter `capture_later = false` or is not set) `id_sale` will be returned.

**`redirect_url` applies only to `redirect_transaction.`

If you perform sale authorization only (by setting `capture_later = true`), `id_sale_authorization` will be returned.

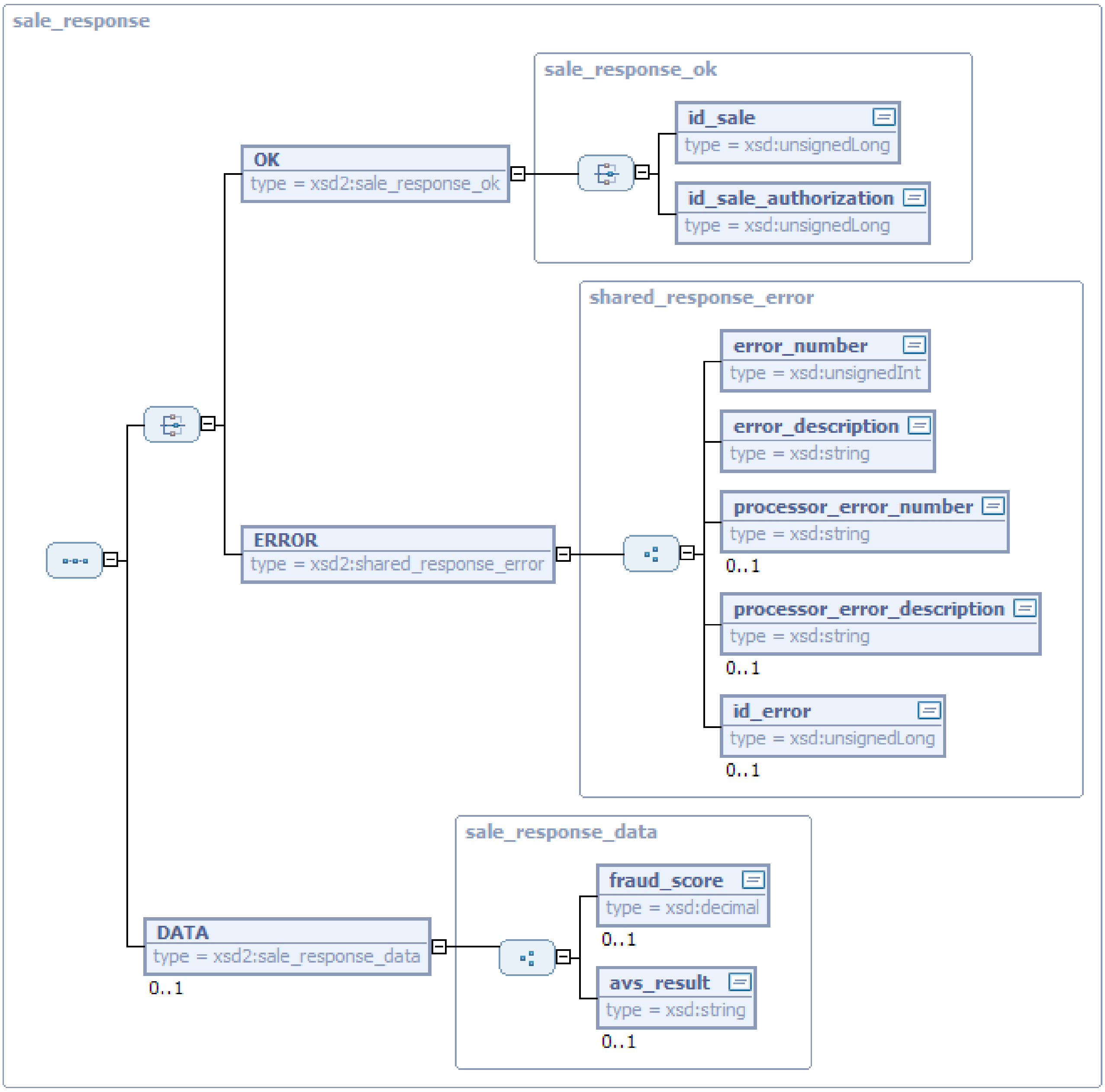
ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.
`processor_error_number`	string	Error number from the processor (acquirer).
`processor_error_description`	string	Textual description of the error from the processor (acquirer).
`id_error`	unsigned long	Identification number of the attempt to make a sale.

DATA object attributes:

Attribute	Format	Description
`fraud_score`	decimal	Fraud score of the transaction. Decimal values from 0 to 10. The lower, the better.
`avs_result`	string	Code returned from Address Verification System (AVS). See Appendix A for all possible values.

multiSale response structure overview:



Error number codes:

Error code	Type	Description
302	recoverable	Direct debit is not accessible for this country [account_country_code].
303	recoverable	Direct debit declined.

312	recoverable	Account holder name is not valid.
313	recoverable	Customer name is not valid.
314	recoverable	Customer e-mail is not valid.
315	recoverable	Customer address (street and house) is not valid.
316	recoverable	Customer city is not valid.
317	recoverable	Customer zip code is not valid.
318	recoverable	Customer state is not valid.
319	recoverable	Customer country is not valid.
320	recoverable	Amount is not valid.
321	recoverable	Amount is too low.
322	recoverable	Currency code is not valid.
323	recoverable	Customer IP address is not valid.
324	recoverable	Description is not valid.
325	recoverable	Account country is not valid.
326	recoverable	Bank code (SWIFT/BIC/BLZ) is not valid.
327	recoverable	Account number is not valid.
401	recoverable	Multiple same transactions lock triggered. Wait [number of seconds] s and try again.
402	recoverable	Payment gateway problem. Please try again later.
403	recoverable	Card declined.
404	recoverable	Transaction in this currency [currency code] is not allowed.
405	recoverable	Unknown payment method or method not set.
406	recoverable	More than one payment method provided. Only one payment method is allowed.
407	recoverable	Capture later not possible with this payment method.
408	recoverable	Feature [feature] not available for this payment method.
409	recoverable	Overriding default [feature] not allowed for this merchant account.
410	recoverable	Unsupported payment method.

411	recoverable	Card number format is not valid.
412	recoverable	Card expiration year is not valid.
413	recoverable	Card expiration month is not valid.
414	recoverable	Card expiration year in the past.
415	recoverable	Card has expired.
416	recoverable	Card code (CVV2/CVC2/CID) format is not valid.
417	recoverable	Name on card is not valid.
418	recoverable	Cardholder name is not valid.
419	recoverable	Cardholder e-mail is not valid.
420	recoverable	Cardholder address (street and house) is not valid.
421	recoverable	Cardholder city is not valid.
422	recoverable	Cardholder zip is not valid.
423	recoverable	Cardholder state is not valid.
424	recoverable	Cardholder country is not valid
425	recoverable	Amount is not valid.
426	recoverable	Amount is too low.
427	recoverable	Currency code is not valid.
428	recoverable	Client IP is not valid.
429	recoverable	Description is not valid.
430	recoverable	Unknown card type or card number invalid.
431	recoverable	Card issue number is not valid.
432	recoverable	Fraud check on is not valid.
433	recoverable	AVS level is not valid.
441	recoverable	Sale Authorization ID is not valid.
442	recoverable	Sale Authorization ID not found or the authorization has been closed.
443	recoverable	Capture sale amount greater than the authorization amount.

488	fatal	Cannot refund this sale.
501	fatal	Internal server error. Please try again later.
502	fatal	Payment gateway error. Please try again later.
503	fatal	Payment method [payment_method_name] not allowed for this merchant account.
505	fatal	This merchant account is inactive.
601	fatal	Fraud attempt detected. Score is: [score] (range is 0-10).
611	fatal	Blacklisted account number found.
612	fatal	Blacklisted card country found.
613	fatal	Blacklisted card number found.
614	fatal	Blacklisted customer country found.
615	fatal	Blacklisted customer email found.
616	fatal	Blacklisted customer IP address found.
731	fatal	Completed authentication with this Secure3d ID not found.
732	fatal	Sale and 3-D Secure card number are different.
733	fatal	Sale and 3-D Secure card expiration year are different.
734	fatal	Sale and 3-D Secure card expiration month are different.
735	fatal	Sale and 3-D Secure amount are different.
736	fatal	Sale and 3-D Secure currency code are different.
737	fatal	Sale with ID [id_sale] was performed for this Secure3d ID.

Returning to the merchant's website:

This applies only to redirect transatnions. If a bank returns an instant notification in response, the customers is redirected to the merchant's website and the following parameters are returned by GET or POST:

- [id_sale]
- [amount]

- [currency]
- [status] - PENDING (when a transaction wasn't confirmed yet) or CLEARED (when a transation was successful and confirmed)
- [hash] – it is counted using the SHA1 function with the following formula:
  <salt> + "|" + <id_sale> + "|" + <status> + "|" + <amount> + "|" + <currency>

The final confirmation is always sent by the PayLane systems separately (using the PayLane notification system) right after payment confirmation. In this case, notifications are extended with two following parameters:

- [expected_amount] – the amount that was expected by the merchant,
- [expected_currency] – expected transaction currency code.

For the complete notification parameter's list, please check out the PayLane Transaction Notifications document.

3.2.2. captureSale

This function will perform funds capture previously authorized using `multiSale` method with a parameter `capture_later = true`. In will create a sale entry in PayLane system.

Request:

Field	Format	Description
`id_sale_authorization`	unsigned long	Identification number of sale authorization returned by `multiSale` method. This parameter is required.
`amount`	decimal (12,2)	Amount to be captured. This amount must not be larger than previously authorized amount. It is possible to capture lower amount than authorized (e.g. when charging for partially fulfilled order).
`description`	string, 2-200 characters	A short description of the sale. May be left empty.

Note: If capture amount is equal to authorized amount, the sale authorization will be automatically closed.

Otherwise, you may use `closeAuthorization` method or leave the authorization to expire (authorizations, whether capture was performed or not, usually expire after 7 calendar days).

Response:

Field	Format	Description
`Response`	An object that will contain one property which will be also an object.	Status of the capture attempt against a sale authorization. An object that contains one of two properties which are also objects: • OK – refund has been completed • ERROR – refund was not performed. The reason will be expressed using the error number and a description. The error codes table is in a separate table below.

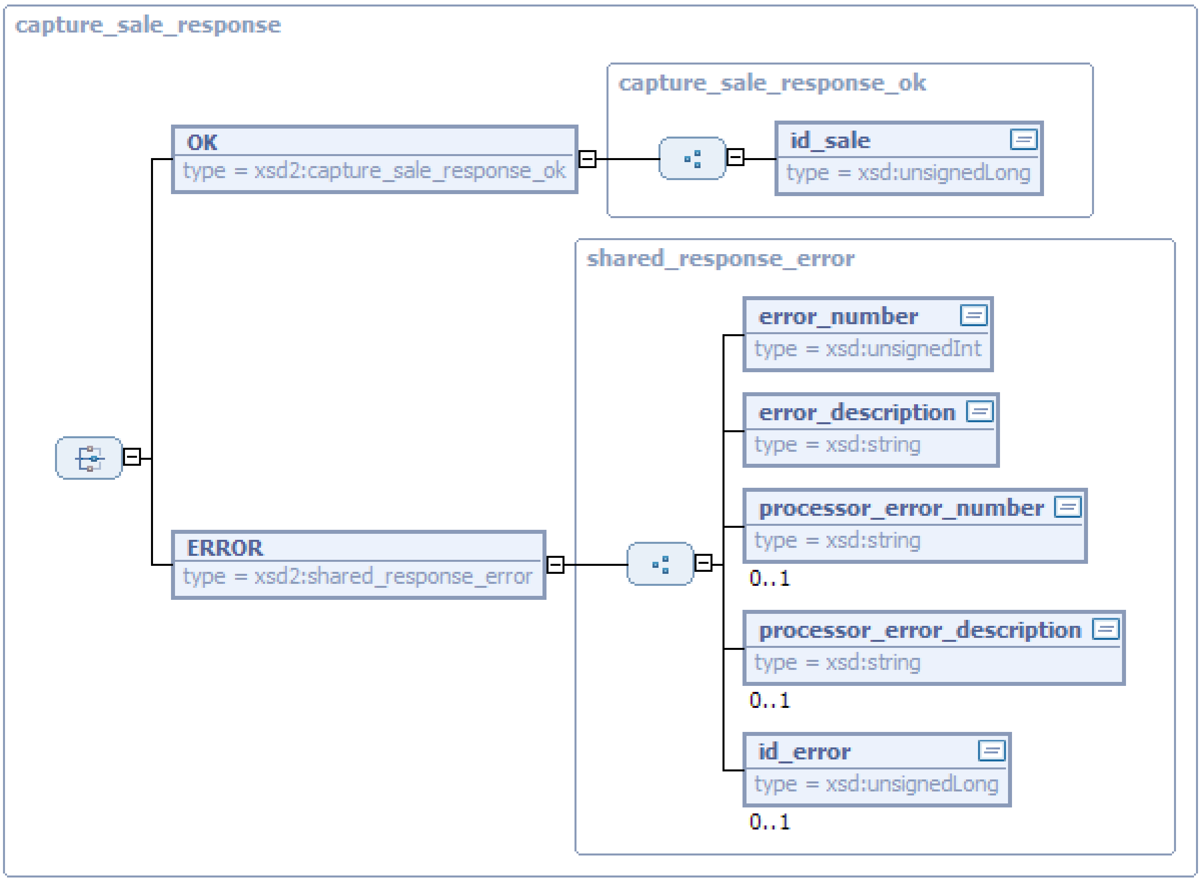
OK object attributes:

Attribute	Format	Description
`id_sale`	unsigned long	Identification number of the sale. This number should be always saved. Sale identification number is unique for each sale.

ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.
`processor_error_number`	string	Error number from the processor (acquirer).
`processor_error_description`	string	Textual description of the error from the processor (acquirer).
`id_error`	unsigned long	Identification number of the attempt to make a sale.

captureSale response structure overview:



Error number codes:

Error code	Type	Description
401	recoverable	Multiple same transactions lock triggered. Wait [number of seconds] s and try again.
441	recoverable	Sale Authorization ID is not valid.
442	recoverable	Sale Authorization ID not found or the authorization has been closed.
443	recoverable	Capture sale amount greater than the authorization amount.
501	fatal	Internal server error. Please try again later.

3.2.3. closeSaleAuthorization

This function will close previously made sale authorization.

Request:

Field	Format	Description
`id_sale_authorization`	unsigned long	Identification number of the sale authorization that should be closed. This sale authorization identification number is returned by the `multiSale` function (when parameter `capture_later = true`). Without this number closing sale authorization is not possible.

Response:

Field	Format	Description
`response`	An object that will contain one property which will be also an object.	Status of the close attempt against a sale authorization. An object that contains one of two properties which are also objects: • OK – closing was successful • ERROR – closing was not performed. The reason will be expressed using the error number and a description. The error codes table is in a separate table below.

OK object attributes:

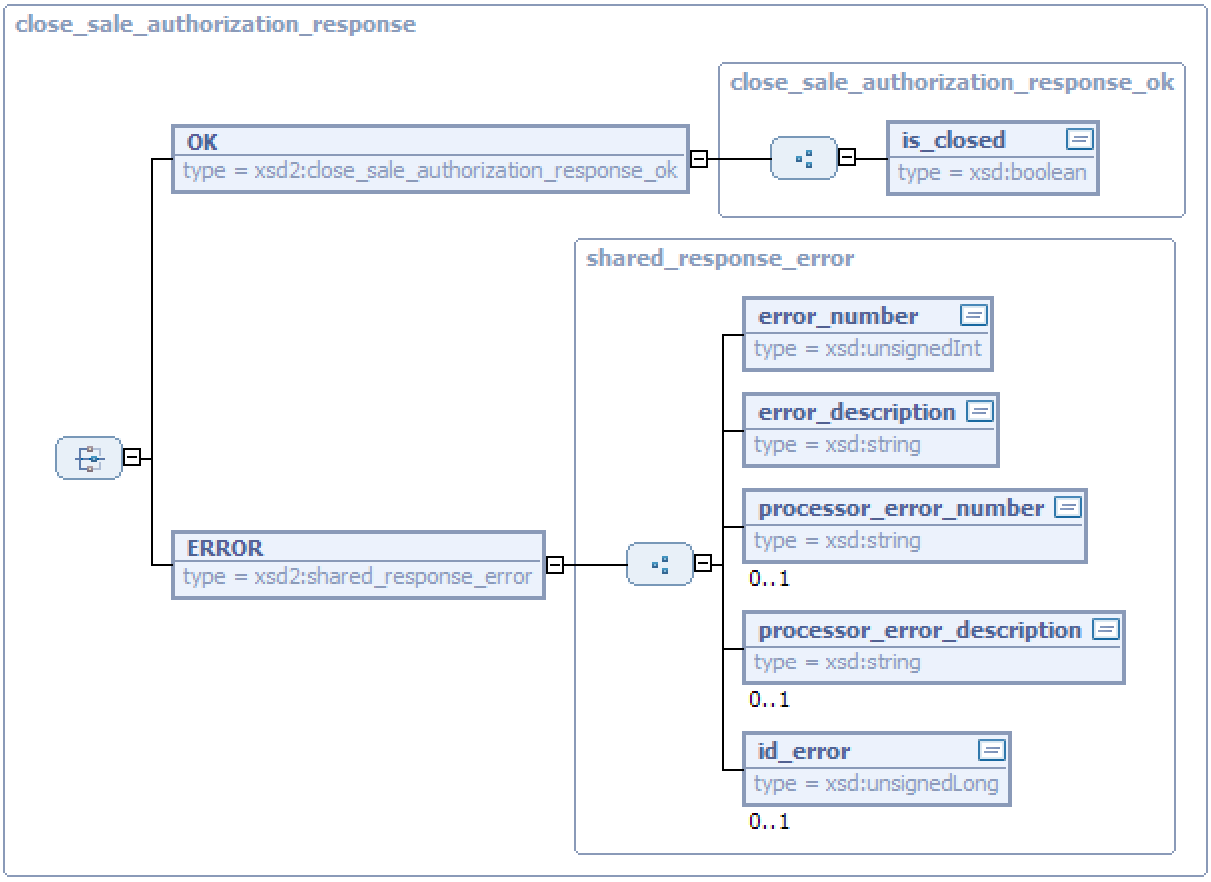
Attribute	Format	Description
`id_closed`	boolean	Indicator that sale authorization was successfully closed. It is always = `true`.

ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.

`processor_error_number`	string	Error number from the processor (acquirer).
`processor_error_description`	string	Textual description of the error from the processor (acquirer).
`id_error`	unsigned long	Identification no. of close authorization attempt.

closeSaleAuthorization response structure overview:



Error number codes:

Error code	Type	Description
`401`	recoverable	Multiple same transactions lock triggered. Wait [number of seconds] s and try again.
`441`	recoverable	Sale Authorization ID is not valid.
`442`	recoverable	Sale Authorization ID not found or the authorization has been closed.
`501`	fatal	Internal server error. Please try again later.

3.2.4. refund

This function refunds a sale.

Request:

Field	Format	Description
`id_sale`	unsigned long	Identification number of the sale that should be refunded. This sale identification number is returned by the sale function. Without this number refund is not possible.
`amount`	decimal (12,2)	Amount to be refunded. Always a positive amount. Note that partial refunds are possible and can be issued up to the amount of the original transaction.
`reason`	string, 2-200 characters	A reason for the refund. Should be always entered, as the refund ratio should be kept on the reasonable low level. PayLane risk management team will occasionally check the reason fields of the merchant when the refund ratio is too high.

Response:

Field	Format	Description
`response`	An object that will contain one property which will be also an object.	Status of the refund attempt against a sale. An object that contains one of two properties which are also objects: • OK – refund has been completed • ERROR – refund was not performed. The reason will be expressed using the error number and a description. The error codes table is in a separate table below.

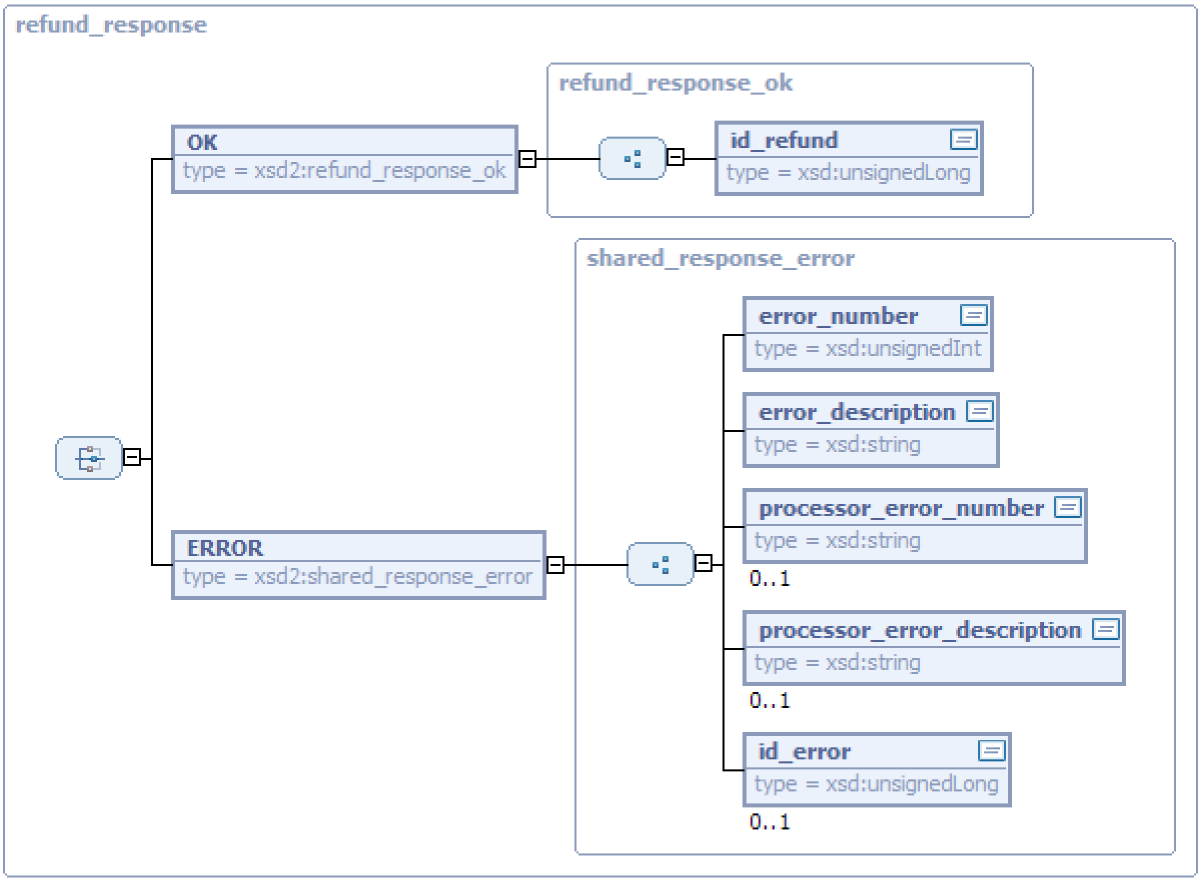
OK object attributes:

Attribute	Format	Description
`id_refund`	unsigned long	Identification number of the refund. This number should be always saved. Refund identification number is unique for each sale.

ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.
`processor_error_number`	string	Error number from the processor (acquirer).
`processor_error_description`	string	Textual description of the error from the processor (acquirer).
`id_error`	unsigned long	Identification number of the attempt to make a refund.

refund response structure overview:



Error number codes:

Error code	Type	Description
401	recoverable	Multiple same transactions lock triggered. Wait [number of seconds] s and try again.
402	recoverable	Payment gateway problem. Please try again later.
481	recoverable	Sale ID is not valid.
482	recoverable	Refund amount is not valid.
483	recoverable	Refund reason is not valid.
484	recoverable	Sale ID not found.

485	recoverable	Cannot refund. Chargeback assigned to Sale ID.
486	recoverable	Cannot refund. Exceeded available refund amount.
487	recoverable	Cannot refund. Sale is already completely refunded.
488	recoverable	Cannot refund this sale.
501	fatal	Internal server error. Please try again later.
502	fatal	Payment gateway error. Please try again later.

3.2.5. resale

This function performs a sale that will use cardholder's data from the sale that was done before. It is usually used to perform recurring billing or charging customers for additional products/services that they order after their initial purchase.

Please note that this function requires card code (CVV2/CVC2/CID) for merchant accounts that are not natively enabled for recurring transactions. The native handling of recurring transactions is determined on the merchant account basis and the decision to allow recurring transactions without supplying card codes is subject to acquirer's approval.

Request:

Field	**Required**	**Format**	**Description**
id_sale	Y	unsigned long	Identification number of the sale that identifies the cardholder. This sale identification number is returned by either sale (first sale) or resale (subsequent sales) function. Without this number resale is not possible.
amount	Y	decimal (12,2)	Amount to be changed.
currency	Y	string, three letter	Currency for this sale.

		code, all uppercase	Format of the currency as in the ISO 4217 standard.
`description`	N	UTF-8 encoded string, 2-200 characters	A textual description of the sale that should briefly tell what the sale is for.
`card_code`	N/Y*	string, 3-4 characters	Card security code (CVV2/CVC2/CID). Optional parameter. Card code has no meaning for Direct Debit resales.
`processing_date`	N	string	Date when the resale is processed.
`resale_by_authorization`	N	boolean	Default = false

* It is required when the merchant account is not directly enabled for recurring transactions.

Response:

Field	**Format**	**Description**
`response`	An object that will contain one property which will be also an object.	Status of the sale attempt. An object that contains one of two properties which are also objects: • OK – refund has been completed • ERROR – refund was not performed. The reason will be expressed using the error number and a description. The error codes table is in a separate table below.

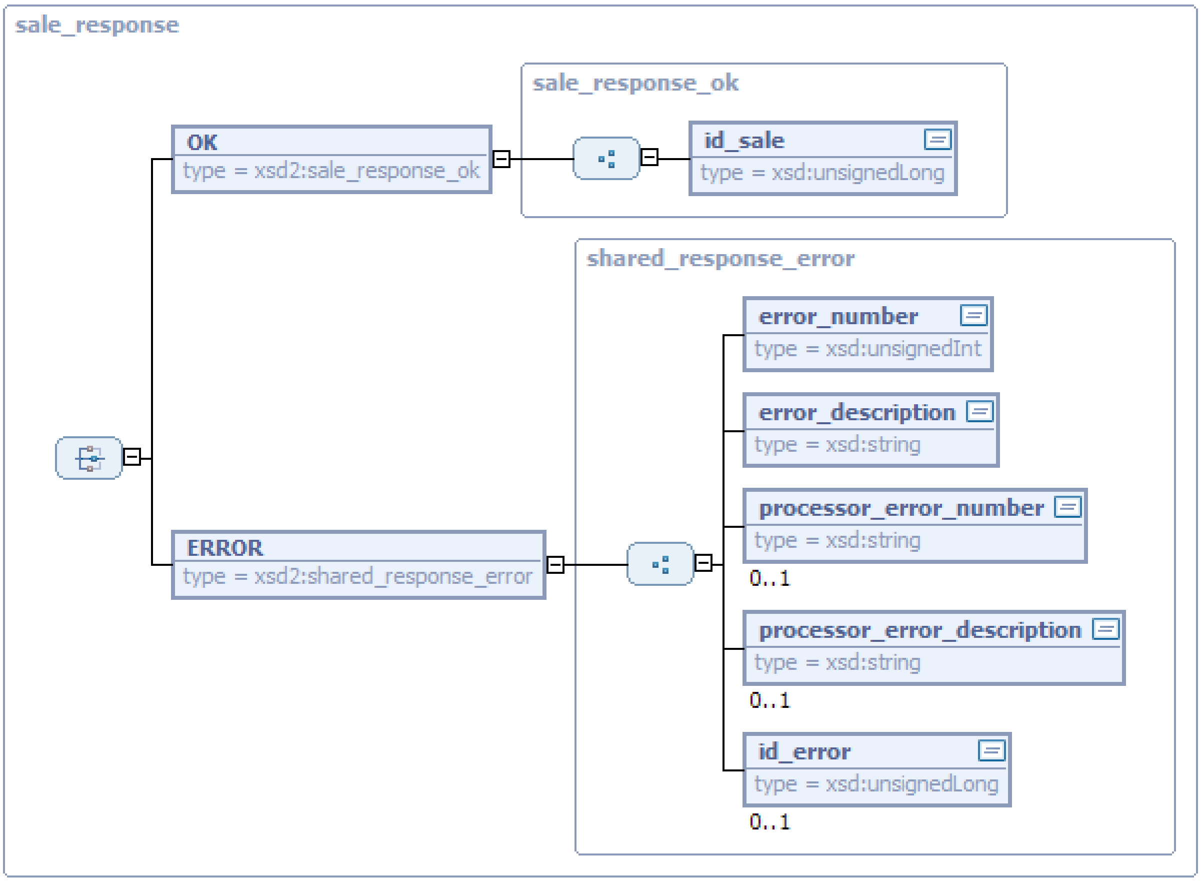
OK object attributes:

Attribute	**Format**	**Description**
`id_sale`	unsigned long	Identification number of the sale. This number should be always saved in case of the refund. Sale identification number is unique for each sale.

ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.
`processor_error_number`	string	Error number from the processor (acquirer).
`processor_error_description`	string	Textual description of the error from the processor (acquirer).
`id_error`	unsigned long	Identification number of the attempt to make a resale.

resale response structure overview:



Error number codes:

Error code	Type	Description
401	recoverable	Multiple same transactions lock triggered. Wait [number of seconds] s and try again.
402	recoverable	Payment gateway problem. Please try again later.
403	recoverable	Card declined.
404	recoverable	Transaction in this currency [currency code] is not allowed.
416	recoverable	Card code (CVV2/CVC2/CID) format is not valid.
470	recoverable	Resale without card code is not allowed for this merchant account.
471	recoverable	Sale ID is not valid.
472	recoverable	Resale amount is not valid.

473	recoverable	Amount is too low
474	recoverable	Resale currency code is not valid.
475	recoverable	Resale description is not valid.
476	recoverable	Sale ID not found.
477	recoverable	Cannot resale. Chargeback assigned to Sale ID.
478	recoverable	Cannot resale this sale.
479	recoverable	Card has expired.
480	recoverable	Cannot resale. Reversal assigned to Sale ID.
501	fatal	Internal server error. Please try again later.
502	fatal	Payment gateway error. Please try again later.

3.2.6. getSaleResult

Request

Field	**Required**	**Format**	**Description**
amount	Y	float	Amount size of performed sale.
description	Y	string	Description of performed sale.

Response:

Case where sale was found:

a) in sales

```
Object
(
    [OK] =>  Object
         (
             [id_sale] - sale ID
         )
)
```

b) in sale errors

```
Object
(
    [OK] =>  Object
        (
            [id_sale_error] - sale error ID
        )
)
```

Case where sale was not found:

sale not found / too many sales found with given request params

```
Object
(
    [ERROR] =>  Object
        (
             [error_number] - error number
             [error_description] - error text description
        )
)
```

3.2.7. checkSales

This function can be used to check the current status of a sale. It should be used for Direct Debit sales where there actual capture of funds is done after the initial pre-authorization (off-line capture of funds). Using this function you can check if the sale has been successfully performed.

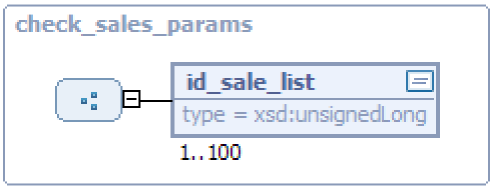
Request:

Field	**Required**	**Format**	**Description**
`check_sale_params`	Y	id_sale_list (xsd:complexType) [1-100 occurrences]	List of Sale IDs for status checking.

id_sale_list structure

Field	Required	Format	Description
`id_sale_list`	Y	unsigned long	Sale ID number.

checkSales request structure overview:



Response:

Field	Format	Description
`response`	An object that will contain one property which will be also an object.	Status of the query attempt. An object that contains one of the two properties which are also objects: • OK – query has been performed • ERROR – query failed – the reason will be expressed using the error number and a description. The error codes table is in a separate table below.

OK object attributes:

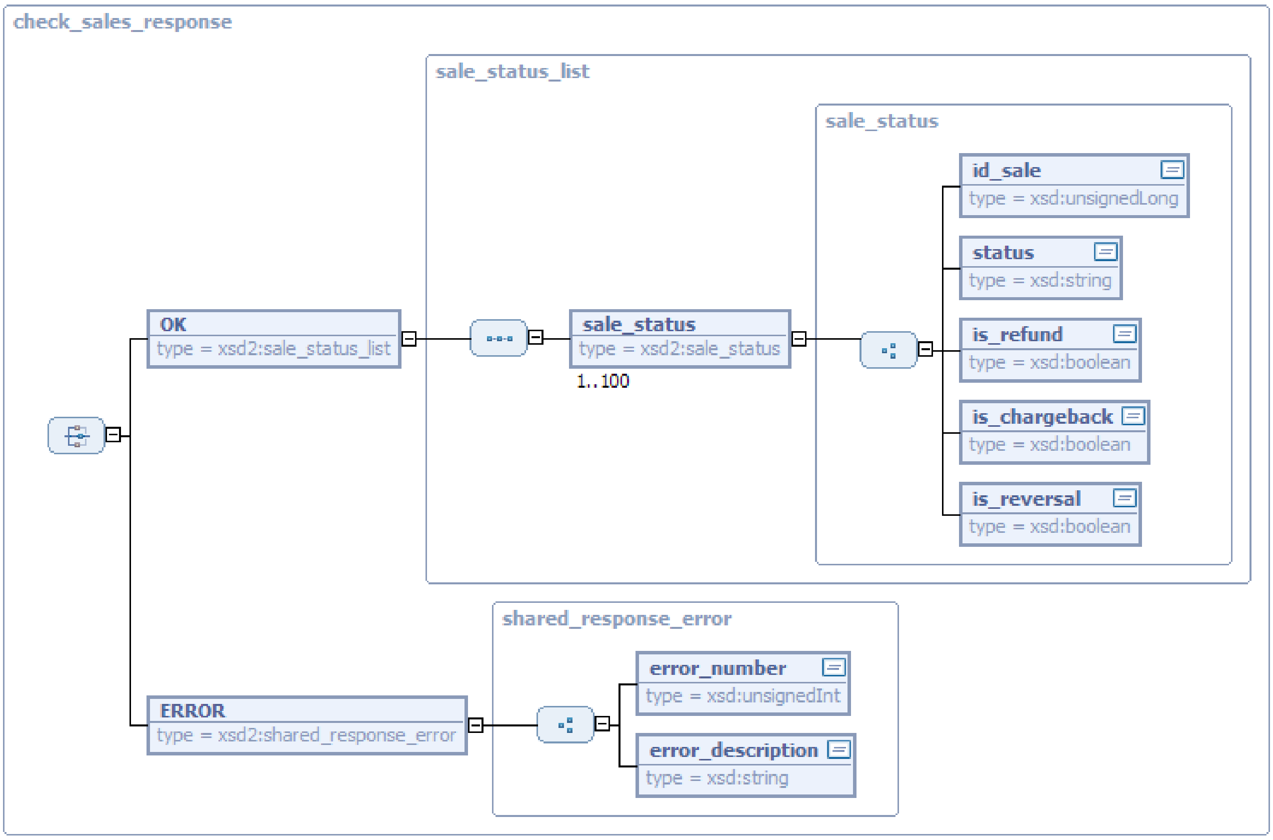
Attribute	Format	Description
`sale_status_list`	sale_status (xsd:complexType) [1-100 occurrencies]	List of status data of checked Sale IDs

sale_status structure explained:

Attribute	Format	Description
`id_sale`	unsigned long	Identification number of the sale. This sale identification number is returned by either sale (first sale) or resale (subsequent sales) function.

`status`	string	Sale status: • NOT_FOUND – Sale ID not found for this merchant account. • PENDING – Sale waiting to be performed – in progress or not completed. • PERFORMED – Sale has been successfully performed. • CLEARED – Sale has been cleared.
`is_refund`	boolean	If true a refund has been performed for this sale.
`is_chargeback`	boolean	If true a chargeback has been performed for this sale.
`is_reversal`	boolean	If true a reversal has been performed for this sale.

checkSales response structure overview:



ERROR object attributes:

Attribute	Format	Description
error_number	unsigned int	Error number. Always three digits.
error_description	string, max 128 chars	Textual description of the error.

Error number codes:

Error code	Type	Description
491	recoverable	Sale ID list is not set or empty.
492	recoverable	Sale ID list is too large (more than 100).
493	recoverable	Sale ID [sale_id] at position [number] is not valid.
501	fatal	Internal server error. Please try again later.

3.2.8. checkCard3DSecureEnrollment

This function is used to check if a card is enrolled in the 3-D Secure program. 3-D Secure Authentication provides additional protection against fraudulent transactions.

Checking card enrollment is a first step in performing 3-D Secure transaction (sale).

3-D Secure enabled transactions require performing additional actions. A full guide to 3-D Secure transaction processing is available in a separate document.

Request:

Field	Required	Format	Description
check_3dsecure_params	Y	structure (xsd:complexType)	Parameters structure.

check_3dsecure_params structure:

Field	Required	Format	Description
s3d_card_data	Y	structure (xsd:complexType)	Card data required to perform the check.
sale_data	Y	structure (xsd:complexType)	Required sale data.
back_url	Y	string	PayLane Secure system will send POST request to this URL after 3-D Secure Authentication indicating success or failure.

s3d_card_data structure:

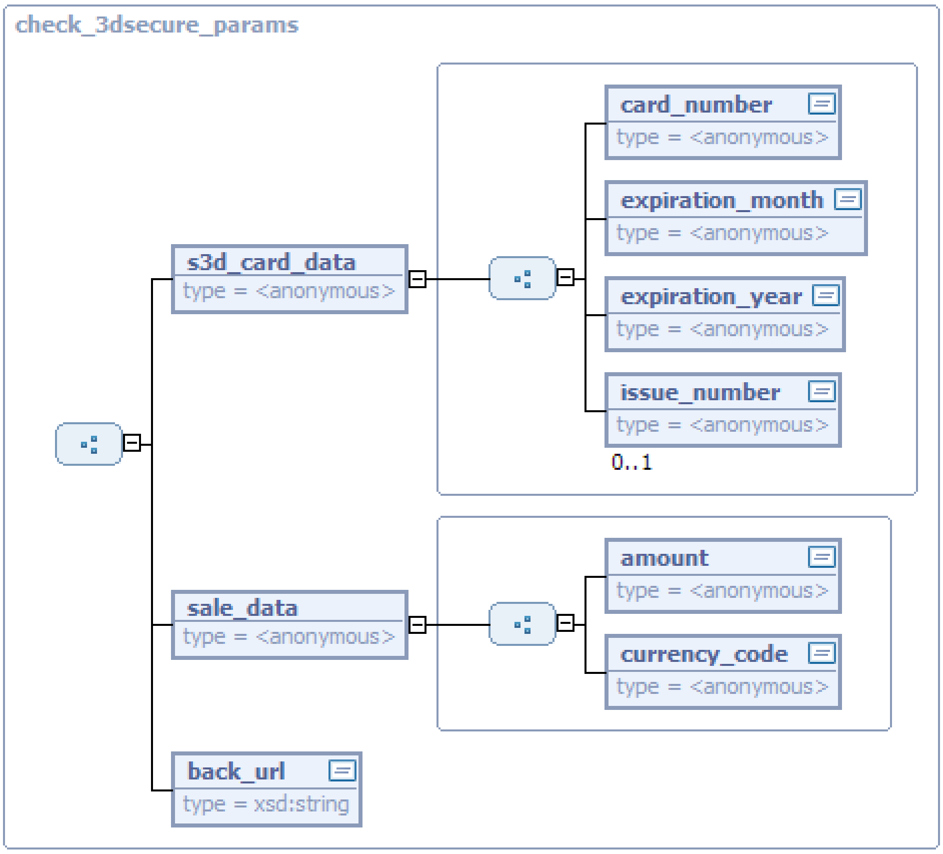
Field	Required	Format	Description
card_number	Y	string, 13-19 characters	The full number of the card without any whitespaces. Only numbers are allowed.
expiration_month	Y	string, 2 characters	Card expiration month, 01-12.
expiration_year	Y	string, 4 characters	Card expiration year, e.g. 2008.
issue_number	N	string, 1-3 characters	Card issue number, present on some cards, e.g. some Maestro UK, Solo. If present, issue number should be set.
card_holder	N*	string, 0-80 characters	Card holder name, e.g. John Smith

* Required only by the Wirecard acquiring bank.

sale_data structure:

Field	Required	Format	Description
amount	Y	decimal (12,2)	Total amount of the sale. Decimal point i a dot. Example: 134.25
currency_code	Y	string (3)	Currency of the sale in ISO 4217 format. All uppercase.

checkCard3DSecureEnrollment request structure overview:



Response:

Field	Format	Description
`response`	An object that will contain one property which will be also an object.	Status of the query attempt. An object that contains one of the two properties which are also objects: • OK – query has been performed • ERROR – query failed – the reason will be expressed using the error number and a description. The error codes table is in a separate table below.

OK object attributes:

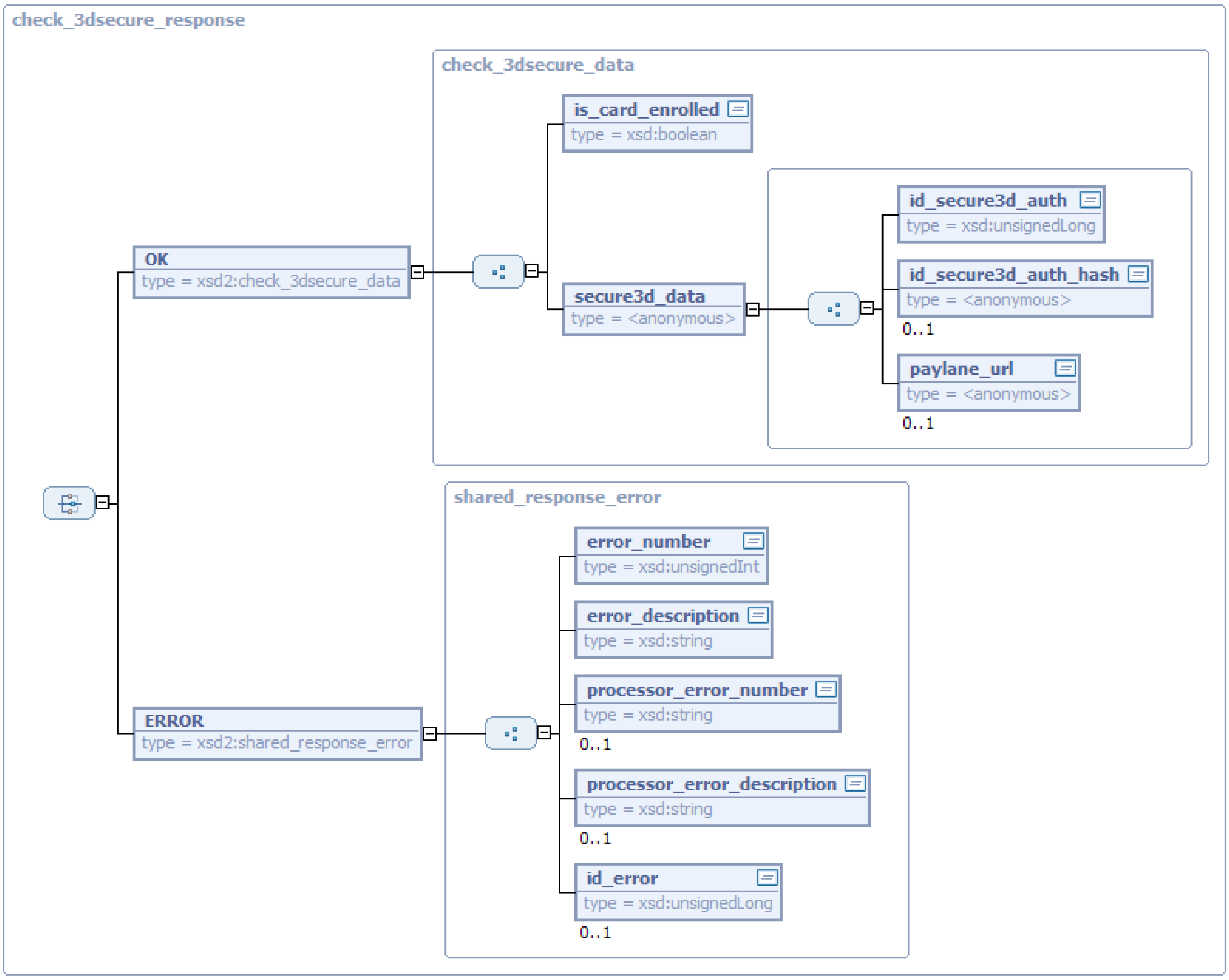
Attribute	Format	Description
`is_card_enrolled`	boolean	Flag indicating if checked card participates in 3-D Secure.
`secure3d_data`	structure (xsd:complexType)	Additional data required to proceed with 3-D Secure Authentication.

secure3d_data structure:

Field	Required	Format	Description
`id_secure3d_auth`	Y	unsigned long	ID of initiated 3-D Secure authentication. It is important to provide this ID when continuing the sale transaction even if card does not participate in 3-D Secure.
`id_secure3d_auth_hash`	N*	string (40)	3-D Secure authentication hash. It is necessary to provide this hash in order to continue 3-D Secure Authentication using PayLane Secure.
`paylane_url`	N*	string max 256 characters	Full link to PayLane Secure including the hash as described above. Performing 3-D Secure Authentication requires redirecting user to this page. More information can be found in separate document.

* Only present when card was successfully checked as participating in 3-D Secure

checkCard3DSecureEnrollment response structure overview:



ERROR object attributes:

Attribute	**Format**	**Description**
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.

Error number codes:

Error code	Type	Description
501	fatal	Internal server error. Please try again later.
504	fatal	Service [service name] not accessible for this merchant account.
505	fatal	This merchant account is inactive.
701	fatal	3-D Secure authentication server error. Please try again later or use card not enrolled in 3-D Secure.
702	recoverable	3-D Secure authentication server problem. Please try again later or use card not enrolled in 3-D Secure.
703	fatal	3-D Secure authentication failed. Credit card cannot be accepted for payment.
704	recoverable	3-D Secure authentication failed. Card declined.
711	recoverable	Card number format is not valid.
712	recoverable	Card expiration year is not valid.
713	recoverable	Card expiration month is not valid.
714	recoverable	Card has expired.
715	recoverable	Amount is not valid.
716	recoverable	Currency code is not valid.
717	recoverable	Back URL is not valid.
718	recoverable	Unknown card type or card number invalid.
719	recoverable	Card issue number is not valid.
720	recoverable	Unable to verify enrollment for 3-D Secure. You can perform a normal payment without 3-D Secure or decline the transaction.

3.2.9. paypalSale

This function is used to initiate or perform PayPal Sale.

Request

Field	Required	Format	Description
currency_code	Y	string	Currency code
amount	Y	decimal	Amount to process
return_url	Y	string	Return url
cancel_url	Y	string	Cancel url
description	Y	string	Transaction description
recurring	N	array	Recurring settings

Recurring structure

Field	Format	Description
start_date	date YYY-MM-DD	Recurring start date
amount	decimal	Amount to recurring
period	string	values: day | week | month | year

Response

```
Object
(
    [OK] => Object
        (
            [id_paypal_checkout] - checkout ID
            [redirect_url] => confirm transaction url
        )

)
```

3.2.10. paypalAuthorization

This method is used to perform sale authorization. It returns id_paypal_checkout and an URL to PayPal website. Operation requires redirecting user to this page.

Request

Field	Required	Format	Description
`currency_code`	Y	string	Currency code
`amount`	Y	decimal	Amount to process
`return_url`	Y	string	Return url
`cancel_url`	Y	string	Cancel url
`description`	Y	string	Transaction description

Response

```
Object
(
    [OK] => Object
        (
            [id_paypal_checkout] - checkout ID
            [redirect_url] => confirm transaction url
        )

)
```

3.2.11. paypalStopRecurring

This method is used to stop recurring profile

Request

Field	Format	Description
`id_paypal_recurring`	int	ID PayPal recurring

Response

```
Object
(
    [OK] => Object
        (
            [id_paypal_recurring] - ID paypal recurring
        )

)
```

3.2.12. paypalGetSaleId

This method is used to get PayLane sale ID by id checkout.

Request

Field	Format	Description
`id_paypal_checkout`	int	ID PayPal checkout

Response

```
Object
(
    [OK] => Object
        (
            [id_sale] - ID sale
        )
)
```

3.2.13. paypalGetSaleAuthorizationId

This method is used to get Paylane sale authorization ID by id checkout.

Request

Field	Format	Description
`id_paypal_checkout`	int	ID PayPal checkout

Response

```
Object
(
    [OK] => Object
         (
             [id_sale_authorization] - ID sale authorization
         )
)
```

3.2.14. checkLastPayPalRecurringSale

This function can be used to check details of the last sale performed as PayPal recurring transaction.

Request

Field	Required	Format	Description
`id_paypal_recurring`	Y	unsigned long	PayPal Recurring ID number

Response:

Field	Format	Description
`response`	An object that will contain one property which will be also an object.	Status of the request. An object that contains one of two properties which are also objects: • OK - there is a sale based on this recurring, • ERROR - sale was not found. The reason

		will be expressed using the error number and a description.

ERROR object attributes:

Attribute	Format	Description
`error_number`	unsigned int	Error number. Always three digits.
`error_description`	string, max 128 chars	Textual description of the error.

OK object attributes:

Attribute	Format	Description
`id_sale`	unsigned long	Identification number of the sale. This number should be always saved in your system to perform other operations on a sale (for example refunds). Sale identification number is unique for each sale.
`date`	date	Date.

Appendix A – Address Verification System (AVS) result codes

Visa, MasterCard, American Express codes

Code	Description
A	Street address matches, but 5-digit and 9-digit postal code do not match.
B	Street address matches, but postal code not verified.
C	Street address and postal code do not match.
D	Street address and postal code match.
E	AVS data is invalid or AVS is not allowed for this card type.
F	Card member's name does not match, but billing postal code matches.
G	Non-U.S. issuing bank does not support AVS.
H	Card member's name does not match. Street address and postal code match.
I	Address not verified.
J	Card member's name, billing address, and postal code match.
K	Card member's name matches but billing address and billing postal code do not match.
L	Card member's name and billing postal code match, but billing address does not match.
M	Street address and postal code match.
N	Street address and postal code do not match.
O	Card member's name and billing address match, but billing postal code does not match.
P	Postal code matches, but street address not verified.
Q	Card member's name, billing address, and postal code match.
R	System unavailable.
S	Bank does not support AVS.
T	Card member's name does not match, but street address matches.

U	Address information unavailable. Returned if the U.S. bank does not support non-U.S. AVS or if the AVS in a U.S. bank is not functioning properly.
V	Card member's name, billing address, and billing postal code match.
W	Street address does not match, but 9-digit postal code matches.
X	Street address and 9-digit postal code match.
Y	Street address and 5-digit postal code match.
Z	Street address does not match, but 5-digit postal code matches.

Maestro UK codes

Code	**Description**
11	Post code and address were not checked.
12	Post code was not checked, address matches.
14	Post code was not checked, address does not match.
18	Post code was not checked, address partially matches.
21	Post code matches, address was not checked.
22	Post code and address match.
24	Post code matches, address does not match.
28	Post code matches, address partially matches.
41	Post code does not match, address was not checked.
42	Post code does not match, address matches.
44	Post code and address do not match.
48	Post code does not match, address partially matches.
81	Post code partially matches, address was not checked.
82	Post code partially matches, address matches.
84	Post code partially matches, address does not match.
88	Post code partially matches, address partially matches.

Appendix B – Test credit card numbers

Number	Vendor	Notes/purpose
		Common sale tests
4111111111111111	Visa	Sale successful.
5500000000000004	MasterCard	Sale successful.
370000000000002	American Express	Sale successful.
6759649826438453	Maestro UK	Sale successful.
6331101999990016	Solo	Sale successful.
5868241608255333338	Maestro International	Sale successful.
		3-D Secure tests
4012001036275556	Visa	Unable to verify card enrollment (enrollment check error 720)
4012001038488884	Visa	Unable to verify card enrollment (enrollment check error 720)
4012001036298889	Visa	Unable to verify card enrollment (enrollment check error 720)
4012001038443335	Visa	3-D Secure Enrollment testing – card not enrolled in 3-D Secure
4012001036853337	Visa	Card enrolled, verification failed (sale error 703)
4012001036983332	Visa	Card enrolled, verification failed (sale error 703)
4012001037490006	Visa	Card enrolled, verification failed (sale error 703)
4012001037167778	Visa	Card not eligible for 3-D Secure, sale successful
4012001037461114	Visa	Card enrolled, authentication failure (sale error 704)
4012001037484447	Visa	Card enrolled, authentication not available (sale error 725)
		AVS (Address Verification System) tests
4055018123456780	Visa	AVS result “X” (exact match)

4055019123456788	Visa	AVS result “Y” (exact match)
4055010123456787	Visa	AVS result “A” (ZIP mismatch)
4055017123456782	Visa	AVS result “W” (street mismatch)
4055020123456786	Visa	AVS result “Z” (street mismatch)
4055013123456781	Visa	AVS result “N” (ZIP and street mismatch)
4055011123456785	Visa	AVS result “E” (logical error/AVS not supported)
4055012123456783	Visa	AVS result “G” (logical error/AVS not supported)
4055015123456786	Visa	AVS result “S” (logical error/AVS not supported)
4055016123456784	Visa	AVS result “U” (logical error/AVS not supported)
4055014123456789	Visa	AVS result “R” (technical error/issuer not available)

Note: Testing 3-D Secure and/or AVS makes sense only if your test account has these mechanisms enabled. Sales will various AVS results will fail with error 403 depending on rejection policy set for the account.

Using different card numbers than listed will universally result in: “card is 3-D Secure enrolled, AVS result “S” (or “11” for Maestro UK / Solo cards), sale successful”.

Any questions? Please contact us:

Phone numbers:
UK phone: +44 2030 514075
Polish phone: +48 58 732 21 11
Fax: +48 58 668 31 46

E-mails:
General: info@paylane.com,
Sales: sales@paylane.com,
For Partners: partners@paylane.com,
Merchant Support: support@paylane.com

United Kingdom:
PayLane UK Ltd
46 Station Road
North Harrow, Middlesex HA2 7SE

Company No. 6493144
VAT ID GB991899439

Poland:
PayLane Sp. z o.o.
Arkonska Business Center
ul. Arkonska 6/A3, 80–387 Gdansk

NIP: 586-214-10-89
Regon: 220010531
KRS: 0000227278
Kapitał zakładowy: 60.000 PLN

Visit and follow us on:
Facebook: http://www.facebook.com/PayLane,
Twitter: http://twitter.com/PayLane,
LinkedIn: http://linkedin.com/company/paylane.