DirectLink

Integration Guide for the Server-to-Server Solution – Version 3.5

www.ogone.com

Contents

1 How Does DirectLink Work? 4

2 General Procedures and Security Settings 5

2.1 Request form 5
2.2 Security 5
2.2.1 Encryption 5
2.2.2 IP address 5
2.3 Response parsing 6

3 Request a New Order 7

3.1 Order request 7
3.1.1 Request URL 7
3.1.2 Request parameters 8
3.1.3 Test page 10
3.2 Order response 10
3.3 Possible response statuses 11
3.4 Duplicate request 12
3.5 Additional security: SHA signature 12

4 Direct Maintenance: Maintenance on Existing Orders 14

4.1 Maintenance request 14
4.1.1 Request URL 14
4.1.2 Request parameters 14
4.1.3 Test page 15
4.2 Maintenance response 15
4.3 Possible transaction statuses 16
4.4 Duplicate request 17

5 Direct Query: Querying the Status of an Order 18

5.1 Query request 18

5.1.1 Request URL 18
5.1.2 Request parameters 18
5.1.3 Test page 18
5.2 Query response 19
5.2.1 Transactions processed with e-Commerce 20
5.3 Possible response statuses 20
5.4 Direct Query as fallback 20

6 Appendix 1: PM Exceptions 22
6.1 Direct Debits 22
6.1.1 Eenmalig-doorlopende machtiging NL 22
6.1.2 ELV - Direct Debits DE 22
6.1.3 Direct Debits AT 23
6.2 PM with only maintenance possible via DirectLink 23

7 Appendix 2: Special Format Travel 24

8 Appendix 3: Troubleshooting 26

9 Appendix 4: List of Parameters to be included in SHA IN Calculation 28

1 How Does DirectLink Work?

DirectLink allows you to set up customized links between your applications and our system, as if our system were simply a local server. It provides program to program (server to server) access between the merchant's software and our payment and administration functions. The merchant's program interacts directly with our remote API without human intervention.

Using DirectLink, there is no contact between our system and the merchant's customer. The merchant transmits all the information required to make the payment directly to our system in an HTTPS post request. Our system requests the financial transaction (synchronously or asynchronously) to the relevant acquirer and returns the response to the merchant in XML format. The merchant's program reads the response and resumes its processing.

The merchant is therefore responsible for collecting and storing sensitive payment details of his customers. He must ensure the confidentiality and security of those details by means of encrypted web communication and server security. If the merchant does not want to store sensitive information such as card numbers, we recommend that he use the Alias option in his account (please refer to the **Alias Manager** integration guide for more information).

The merchant can process new orders, perform maintenance on existing orders and query the status of an order using DirectLink.

Even if the merchant has automated requests with DirectLink, he can view the history of the transaction in the back-office manually using his web browser or a report download. For the configuration and functionality of the administration site, please refer to the **Back-Office User Guide**.

2 General Procedures and Security Settings

IMPORTANT: The following general procedures and security controls are valid for all DirectLink requests: new order requests, maintenance requests and direct queries.

2.1 Request form

For new order requests, maintenance requests and direct queries, the merchant must send requests with certain parameters to specific URLs. The payment/maintenance/query parameters must be sent in a POST request as follows:

PSPID=value1&USERID=value2&PSWD=value3&…

The type/subtype indicating the Media Type in the Content-Type entity-header field in the POST request needs to be "application/x-www-form-urlencoded".

DirectLink works in "one request-one reply" mode, each payment is processed individually. Our system handles individual transaction requests via DirectLink and can work synchronously (where this option is technically supported), i.e. we wait for the bank's reply before returning an XML response to the request.

2.2 Security

When we receive a request on our servers, we check the level of encryption and the IP address which the request was sent from.

2.2.1 Encryption

DirectLink is built on a robust and secure communication protocol. DirectLink API is a set of instructions submitted with standard HTTPS Post requests.

At the server end, we use a certificate delivered by Verisign. The SSL encryption guarantees that it is *our* servers you are communicating with and that your data is transmitted in encrypted form. There is no need for a client SSL certificate.

When we receive a request, we check the level of encryption. We only allow the merchant to connect to us in secure https mode using SSL v3. This guarantees 128-bit encryption.

2.2.2 IP address

For each request, our system checks the IP address from which the request originates to ensure the requests are being sent from the merchant's server. In the IP address field of the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page of your account you must enter the IP address(es) or IP address range(s) of the servers that send your requests .

If the IP address from which the request originates has not been declared in the IP address field of the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page in your account, you will receive the error message *"unknown order/1/i".* The IP address the request was sent from will also be displayed in the error message.

2.3 Response parsing

We will return an XML response to your request. Please ensure that your systems parse this XML response as tolerantly as possible to avoid issues in the future, e.g. avoid case sensitive attribute names, do not prescribe a specific order for the attributes returned in responses, ensure that new attributes in the response will not cause issues etc…

3 Request a New Order

3.1 Order request

Please refer to Chapter 2 for the General Procedures and Security Controls.

3.1.1 Request URL

The request URL in the TEST environment is https://secure.ogone.com/ncol/test/orderdirect.asp.

The request URL in the PRODUCTION environment is https://secure.ogone.com/ncol/prod/orderdirect.asp.

IMPORTANT: Do not forget to replace "test" with "prod" in the request URL when you switch to your PRODUCTION account. If you forget to change the request URL, once you start in production with real orders, your transactions will be sent to the test environment and will not be sent to the acquirers/banks.

3.1.2 Request parameters

The following table contains the request parameters for sending a new order:

Parameter	Usage
PSPID	Your affiliation name in our system.
OrderID	Your unique order number (merchant reference).
USERID	Name of your application (API) user. Please refer to the User Manager documentation for information on how to create an API user.
PSWD	Password of the API user (USERID).
amount	Amount to be paid MULTIPLIED BY 100 since the format of the amount must not contain any decimals or other separators.
currency	ISO alpha order currency code, for example: EUR, USD, GBP, CHF, …
CARDNO	Card/account number.
ED	Expiry date (MM/YY or MMYY).
COM	Order description.
CN	Customer name.
EMAIL	Customer's email address.
SHASign	Signature (hashed string) to authenticate the data (see section 3.5).
CVC	Card Verification Code. Depending on the card brand, the verification code will be a 3- or 4-digit code on the front or rear of the card, an issue number, a start date or a date of birth (for more information, please refer to https://secure.ogone.com/ncol/card_verification_code1.asp)
Ecom_Payment_Card_Verification	Same as CVC.
Owneraddress	Customer's street name and number.
OwnerZip	Customer's ZIP code.
ownertown	Customer's town/city name.
ownercty	Customer's country, e.g. BE, NL, FR, …
ownertelno	Customer's telephone number.
BRAND	Card brand.

Operation	Defines the type of requested transaction. You can configure a default operation (payment procedure) in the "Global transaction parameters" tab, "Default operation code" section of the Technical Information page. When you send an operation value in the request, this will overwrite the default value. Possible values: ▪ **RES**: request for authorization ▪ **SAL**: request for direct sale ▪ **RFD**: refund, not linked to a previous payment, so not a maintenance operation on an existing transaction (you can not use this operation without specific permission from your acquirer).
GloborderID	Reference grouping several orders together; allows you to request a joint maintenance operation on all these transactions at a later time.
Withroot	Adds a root element to our XML response. Possible values: 'Y' or empty.
REMOTE_ADDR	IP address of the customer (for Fraud Detection Module only). If a country check does not need to be performed on the IP address, send 'NONE'.
RTIMEOUT	Request timeout for the transaction (in seconds, value between 30 and 90) **IMPORTANT**: The value you set here must be smaller than the timeout value in your system!
ECI	Electronic Commerce Indicator. You can configure a default ECI value in the "Global transaction parameters" tab, "Default ECI value" section of the Technical Information page. When you send an ECI value in the request, this will overwrite the default ECI value. Possible (numeric) values: 0 - Swiped 1 - Manually keyed (MOTO) (card not present) 2 - Recurring (from MOTO) 3 - Instalment payments 4 - Manually keyed, card present 7 - E-commerce with SSL encryption 9 - Recurring (from e-commerce)

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

The list of possible parameters to send can be longer for merchants who have activated certain options/functionalities in their accounts. Please refer to the respective option documentation for more information on extra parameters linked to the option.

The following request parameters are mandatory in new orders:

- PSPID and USERID
- PSWD
- orderID
- amount(x 100)
- currency
- CARDNO
- ED
- CVC
- operation (not strictly required, but strongly recommended).

3.1.3 Test page

A test page for an order request can be found at https://secure.ogone.com/ncol/test/testodl.asp

3.2 Order response

Our server returns an XML response to the request:

Example of an XML response to an order request:

```
<?xml version="1.0"?>
<ncresponse orderID="99999" PAYID="1111111" NCSTATUS="0" NCERROR="" NCERRORPLUS=""
ACCEPTANCE="12345" STATUS="5" ECI="7" amount="125" currency="EUR" PM="CreditCard"
BRAND="VISA"/>
```

The following table contains a list of the ncresponse tag attributes:

Field	Usage
orderID	Your order reference.
PAYID	Payment reference in our system.
NCSTATUS	First digit of NCERROR.
NCERROR	Error code.
NCERRORPLUS	Explanation of the error code.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
ECI	Electronic Commerce Indicator.
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

The attribute list may be longer for merchants who have activated certain options (e.g. Fraud Detection Module) in their accounts. Please refer to the respective option documentation for further information about additional response attributes linked to the option.

3.3 Possible response statuses

Status	NCERROR	NCSTATUS	Explanation
5 Authorized	0	0	The authorization has been accepted. An authorization code is available in the field "ACCEPTANCE". The status will be 5 if you have configured "Authorisation" as default operation code in your Technical Information page or if you send Operation code RES in your transaction request.
9 Payment requested	0	0	The payment has been accepted. An authorization code is available in the field "ACCEPTANCE". The status will be 9 if you have configured "Sale" as default operation code in your Technical Information page or if you send Operation code SAL in your transaction request.
0 Invalid or incomplete	500....	5	At least one of the payment data fields is invalid or missing. The NCERROR and NCERRORPLUS fields contains an explanation of the error (list available at http://www.ogone.com/ncol/paymentinfos1.asp). After correcting the error, the customer can retry the authorization process.
2 Authorization refused	300....	3	The authorization has been declined by the financial institution. The customer can retry the authorization process after selecting a different payment method (or card brand).
51 Authorization waiting	0	0	The authorization will be processed offline. This is the standard response if you have chosen offline processing in the account configuration. The status will be 51 in two cases: • You have defined "Always offline" in the "Global transaction parameters" tab, "Processing for individual transactions" section of the Technical Information page in your account. • In case the online acquiring system is unavailable and you have defined "Online but switch to offline in intervals when the online acquiring system is unavailable." in the "Global transaction parameters" tab, "Processing for individual transactions" section of the Technical Information page in your account. You cannot retry the authorization process because the payment might be accepted offline.

52 Authorization not known Or 92 Payment uncertain	200...	2	A technical problem arose during the authorization/payment process, giving an unpredictable result. The merchant can contact the acquirer helpdesk to establish the precise status of the authorization/payment or wait until we have updated the status in our system. The customer should not retry the authorization process since the authorization/payment might already have been accepted.

3.4 Duplicate request

If you request processing for an already existing (and correctly treated) orderID, our XML response will contain the PAYID corresponding to the already existing orderID, the ACCEPTANCE given by the acquirer in the previous processing, STATUS "0" and NCERROR "50001113".

3.5 Additional security: SHA signature

The secure requests and inspection of the originating IP address (described in Section 2.2) are primary measures to secure your order/request data. If you wish, you may, however, also include an additional check of the request data to ensure the data our system received are identical to those you sent for your new order. You can use the SHA signature to this end.

The SHA signature is based on the principle of the merchant's server generating a unique character string for each order, hashed with the SHA-1, SHA-256 or SHA-512 algorithms. The result of this hash is then sent to us in the merchant's order request. Our system reconstructs this signature to check the data integrity of the order information sent to us in the request.

This string is constructed by concatenating the values of the fields sent with the order (sorted alphabetically, in the format 'parameter=value'), separated by a passphrase. The passphrase is defined in the Merchant's *Technical information*, under the tab "Data and Origin Verification", section "Checks for DirectLink." For the full list of parameters to include in the SHA Digest, please refer to Appendix 4. Please note that these values are all case sensitive when compiled to form the string before the hash!

IMPORTANT

- all parameter names should be in UPPERCASE (to avoid any case confusion)
- Parameters that do not have a value should NOT be included in the string to hash

When you hash the string composed with the SHA algorithm, a hexadecimal digest will be returned The length of the SHA Digest is 40 characters for SHA-1, 64 for SHA-256 and 128 for SHA-512. This result should be sent to our system in your order request, using the "SHASign" field.

Our system will recompose the SHA string based on the received parameters and compare the Merchant's Digest with our generated Digest. If the result is not identical, the order will be declined. This check ensures the accuracy and integrity of the order data.

You can test your SHASign at https://secure.ogone.com/ncol/test/testsha.asp

Example of a basic SHA-1-IN calculation

parameters (in alphabetical order)
amount: 15.00 -> 1500
CARDNO: 4111111111111111
currency: EUR
Operation: RES
orderID: 1234
PSPID: MyPSPID

SHA Passphrase (In technical info)
Mysecretsig1875!?

string to hash
AMOUNT=1500Mysecretsig1875!?CARDNO=4111111111111111Mysecretsig1875!?CURRENCY=EUR Mysecretsig1875!?OPERATION=RESMysecretsig1875!?ORDERID=1234Mysecretsig1875!?PSPID=My PSPIDMysecretsig1875!?

resulting Digest (SHA-1)
2B459D4D3AF0C678695AE77EE5BF0C83CA6F0AD8

If the SHASign sent in your request does not match the SHASign which we derived using the details of the order and the passphrase entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page, you will receive the error message *"unknown order/1/s".*

If the "SHASign" field in your request is empty but a passphrase has been entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page (indicating you want to use a SHA signature with each transaction), you will receive the error message *"unknown order/0/s".*

4 Direct Maintenance: Maintenance on Existing Orders

A direct maintenance request from your application allows you to: perform a data capture (payment) of an authorized order automatically (as opposed to manually in the back-office); cancel an authorization on an order; renew an authorization of an order; or refund a paid order.

Data captures, authorization cancellations and authorization renewals are specifically for merchants who have configured their account/requests to perform the authorization and the data capture in two steps.

4.1 Maintenance request

Please refer to Chapter 2 for the General Procedures and Security Controls.

4.1.1 Request URL

The request URL in the TEST environment is https://secure.ogone.com/ncol/test/maintenancedirect.asp.

The request URL in the PRODUCTION environment is https://secure.ogone.com/ncol/prod/maintenancedirect.asp.

IMPORTANT: Do not forget to replace "test" with "prod" in the request URL when you switch to your PRODUCTION account. If you forget to change the request URL, once you start in production with real orders, your maintenance transactions will be sent to the test environment and will not be sent to the acquirers/banks.

4.1.2 Request parameters

The following table contains the mandatory request parameters for performing a maintenance operation:

Field	Usage
PSPID	Login details: PSPID and (API) USERID with the USERID's password
USERID	
PSWD	
PAYID	You can send the PAYID or the orderID to identify the original order. We recommend the use of the PAYID.
orderID	

amount	Order amount multiplied by 100. Is only required when the amount of the maintenance differs from the amount of the original authorization. However, we recommend its use in all cases. Our system will check if the maintenance transaction amount is not too high in comparison with the amount of the authorization/payment.
Operation	Possible values: ▪ **REN**: renewal of authorization, if the original authorization is no longer valid. ▪ **DEL**: delete authorization, leaving the transaction open for possible further maintenance operations. ▪ **DES**: delete authorization, closing the transaction after this operation. ▪ **SAL**: partial data capture (payment), leaving the transaction open for a possible other data capture. ▪ **SAS**: (last) partial or full data capture (payment), closing the transaction (for further data captures) after this data capture. ▪ **RFD**: partial refund (on a paid order), leaving the transaction open for a possible other refund. ▪ **RFS**: (last) partial or full refund (on a paid order), closing the transaction after this refund. Please note with DEL and DES that not all acquirers support the deletion of an authorization. If your acquirer does not support DEL/DES, we will nevertheless simulate the deletion of the authorization in the back-office.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

4.1.3 Test page

An example (test page) of a direct maintenance request can be found at: https://secure.ogone.com/ncol/test/testdm.asp.

4.2 Maintenance response

Our server returns an XML response to the request:

Example of an XML response to a direct maintenance request:

```
<?xml version="1.0"?>
<ncresponse orderID="99999" PAYID="1111111" PAYIDSUB="3" NCSTATUS="0" NCERROR=""
NCERRORPLUS="" ACCEPTANCE="12345" STATUS="91" amount="125" currency="EUR" PM="CreditCard"
BRAND="VISA"/>
```

The following table contains a list of the ncresponse tag attributes:

Field	Usage
orderID	Your order reference.
PAYID	Payment reference in our system.
PAYIDSUB	The history level ID of the maintenance operation on the PAYID.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
NCERROR	Error code.
NCSTATUS	First digit of NCERROR.
NCERRORPLUS	Explanation of the error code.
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

The standard ncresponse tag attributes are the same as those for the XML reply to a new order, except for the extra attribute PAYIDSUB.

4.3 Possible transaction statuses

The maintenance orders are always treated **offline** (except for authorization renewals).

Status	NCERROR	NCSTATUS	Explanation
0 Invalid or incomplete	500....	5	At least one of the payment data fields is invalid or missing. The NCERROR and NCERRORPLUS fields give an explanation of the error (list available at http://www.ogone.com/ncol/paymentinfos1.asp).
91 Payment processing	0	0	The data capture will be processed offline.
61- Author. deletion waiting	0	0	The authorization deletion will be processed offline.
92 Payment uncertain	200...	2	A technical problem arose during the payment process, giving an unpredictable result. The merchant can contact the acquirer helpdesk to establish the precise status of the payment or wait until we have updated the status in our system. You should not retry the payment process since the payment might already have been accepted.
62 - Author. deletion uncertain	200...	2	A technical problem arose during the authorization deletion process, giving an unpredictable result. The merchant can contact the acquirer helpdesk to establish the precise status of the payment or wait

			until we have updated the status in our system.
93 - Payment refused	300....	3	A technical problem arose.
63 - Author. deletion refused	300....	3	A technical problem arose.

4.4 Duplicate request

If a maintenance is requested twice for the same order, the second one will theoretically be declined with an error "50001127" (this order is not authorized), because the initial successful transaction will have changed the order status.

5 Direct Query: Querying the Status of an Order

A direct query request from your application allows you to query the status of an order automatically (as opposed to manually in the back-office). You can only query one payment at a time, and will only receive a limited amount of information about the order.

If you need more details about the order, you can look up the transaction in the back-office or perform a manual or automatic file download (please refer to the **Back-Office** User Guide and the **Advanced Batch** Integration Guide).

5.1 Query request

Please refer to Chapter 2 for the General Procedures and Security Controls.

5.1.1 Request URL

The request URL in the TEST environment is https://secure.ogone.com/ncol/test/querydirect.asp.

The request URL in the PRODUCTION environment is https://secure.ogone.com/ncol/prod/querydirect.asp.

IMPORTANT: Do not forget to replace "test" with "prod" in the request URL when you switch to your PRODUCTION account.

5.1.2 Request parameters

The following table contains the mandatory request parameters to perform a direct query:

Field	Usage
PSPID	Login details: PSPID and (API) USERID with the USERID's password
USERID	
PSWD	
PAYID	You can send the PAYID or the orderID to identify the original order. We recommend the use of the PAYID.
orderID	
PAYIDSUB	You can indicate the history level ID if you use the PAYID to identify the original order (optional).

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

5.1.3 Test page

An example (test page) of a direct query request, can be found at: https://secure.ogone.com/ncol/test/testdq.asp.

5.2 Query response

Our server returns an XML response to the request:

Example of an XML response to a direct query:

```
<?xml version=”1.0”?>
<ncresponse orderID=”99999” PAYID=”1111111” PAYIDSUB=”3” NCSTATUS=”0” NCERROR=””
NCERRORPLUS=”” ACCEPTANCE=”12345” STATUS="9" ECI=”7” amount="125" currency="EUR"
PM="CreditCard" BRAND="VISA" CARDNO="XXXXXXXXXXXX1111" IP="212.33.102.55"/>
```

The following table contains a list of the ncresponse tag attributes:

Field	Usage
orderID	Your order reference.
PAYID	Payment reference in our system.
PAYIDSUB	The history level ID of the maintenance operation on the PAYID.
NCSTATUS	First digit of NCERROR.
NCERROR	Error code.
NCERRORPLUS	Explanation of the error code.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
ECI	Electronic Commerce Indicator
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.
CARDNO	The masked credit card number.
IP	Customer’s IP address, as detected by our system in a 3-tier integration, or sent to us by the merchant in a 2-tier integration.

For further technical details about these fields, please refer to the online ***Parameter Cookbook*****.**

The standard ncresponse tag attributes are identical to those for the XML reply to a new order, except for the additional attributes PAYIDSUB, CARDNO and IP.

The attribute list may be longer for merchants who have activated certain options (e.g. the Fraud Detection Module) in their accounts. Please refer to the respective option documentation for more information on extra response attributes linked to the option.

5.2.1 Transactions processed with e-Commerce

If the transaction you want to look up the status for was processed with e-Commerce, you will also receive the following additional attributes (providing you sent these fields with the original e-Commerce transaction).

Field	Usage
complus	A value you wanted to have returned.
(paramplus content)	The parameters and their values you wanted to have returned.

*For further technical details about these fields, please refer to the **Advanced e-Commerce Integration Guide**.*

Example of an XML response to a direct query for an e-Commerce transaction:

<?xml version="1.0"?>
<ncresponse orderID="99999" PAYID="1111111" PAYIDSUB="3" NCSTATUS="0" NCERROR=""
NCERRORPLUS="" ACCEPTANCE="12345" STATUS="9" amount="125" currency="EUR" PM="CreditCard"
BRAND="VISA" CARDNO="XXXXXXXXXXXX1111" IP="212.33.102.55"
COMPLUS="123456789123456789123456789" SessionID="126548354"
ShopperID="73541312"/>

5.3 Possible response statuses

The STATUS field will contain the status of the transaction. For a full list of statuses, please refer to: http://www.ogone.com/ncol/paymentinfos1.asp.

Only the following status is specifically related to the query itself:

Status	NCERROR	NCSTATUS	Explanation
88			The query on querydirect.asp failed.

5.4 Direct Query as fallback

The response times for a DirectLink transaction request are generally a few seconds; some acquirers may, however, have longer response times. If you want to install a check mechanism to verify that our system is up and running smoothly, we suggest you set the request timeout in orderdirect.asp to 30 seconds (30-40 for Diners).

If you have not received a response from our system after 30 seconds, you send a request to querydirect.asp, asking for the status of the transaction you just sent to orderdirect.asp. If you receive an immediate reply containing a non-final status for the transaction, there might be issues at the acquirer's end.

If you have not received an answer to this direct query request after 10 seconds, there might be issues at our end. You can repeat this request to querydirect.asp every 30 seconds until you see you receive a response within 10 seconds.

Please note:

1. This check system will only be able to pinpoint issues at our end if there is also a check at your end to verify that requests are leaving your servers correctly.

2. An issue at our end will not always necessarily be caused by downtime, but could also be as a result of slow response times due to database issues for example.

3. Please use these checks judiciously to avoid bombardment of our servers with requests, otherwise we might have to cut your access to the querydirect.asp page.

IMPORTANT: To protect our system from unnecessary overloads, we prohibit system-up checks comprising the sending of fake transactions or systematic queries, as well as systematic queries to obtain transaction feedback for each transaction.

6 Appendix 1: PM Exceptions

For certain payment methods, the parameter values differ from the standard credit card values.

6.1 Direct Debits

6.1.1 Eenmalig-doorlopende machtiging NL

The following table contains the specific parameter values allowing the transmission of Direct Debits NL transactions via DirectLink.

Field	Value
PM	"Direct Debits NL"
CARDNO	Bank account number. This should always be 10 digits: if the account is less than 10 digits, left pad with zeroes. For PostBank accounts: "000" + 7 digits or "P00" + 7 digits.
OPERATION	Possible values: ▪ **VEN**: debit money from the bank account ▪ **RFD**: credit money to the bank account (maintenance operation)
CN	Bank accountholder's name.
OWNERADDRESS	City of the bank accountholder.
ED	"99/99" or "9999".

6.1.2 ELV - Direct Debits DE

The following table contains the specific parameter values allowing the transmission of ELV transactions via DirectLink.

Field	Value
PM	"Direct Debits DE"
CARDNO	Bank account number. Format: XXXXXXXXXBLZYYYYYYYY XXXXXXXXXX: account number, numeric, 1 to 10 digits. YYYYYYYY: Bank code (Bankleitzahl), 8 digits.
OPERATION	Possible values: ▪ **RES**: authorization ▪ **VEN**: debit money from the bank account
ED	"99/99" or "9999"

6.1.3 Direct Debits AT

The following table contains the specific parameter values allowing the transmission of Direct Debits AT transactions via DirectLink.

Field	Value
PM	"Direct Debits AT"
CARDNO	Bank account number. Format: XXXXXXXXXXXBLZYYYYY XXXXXXXXXXX: account number, numeric, 11 digits. YYYYY: Bank code (Bankleitzahl), 5 digits.
OPERATION	Possible values: ▪ **RES**: authorization ▪ **VEN**: debit money from the bank account
ED	"99/99" or "9999"

6.2 PM with only maintenance possible via DirectLink

For certain (non credit card) payment methods, you cannot send new transactions via DirectLink, but you can send certain maintenance operations via DirectLink. This is the case for: **PostFinance Card**, **PostFinance e-finance**, **PAYPAL Express Checkout** and **TUNZ**. When sending maintenance operations, PM/BRAND/CARDNO/ED are not required data, so no specific values need to be sent for these payment methods.

7 Appendix 2: Special Format Travel

You can send additional data for travel transactions if your acquirer is able to receive and process the data. The following table contains the possible additional fields for travel.

IMPORTANT: The detailed specifications for each field, especially "mandatory/optional", are only mentioned for information purposes and may differ slightly from one acquirer to the other. Also, not all acquirers accept all fields.

Name	Usage		Field details
DataType	"TRAVEL"	mandatory	TRAVEL
AIAIRNAME	Airline name.	optional	max.20
AITINUM	Ticket number Air+ defines this zone as follows: 3 digits for airline prefix (filled with 0's if ticket type <> BSP + 10 chars for ticket number). Other acquirers do not split this zone – it is just the ticket number.	mandatory	max.16
AITIDATE	Ticket issue date. The default value is the transaction date.	optional	MM/DD/YYYY or YYYYMMDD
AICONJTI	Conjunction ticket.	optional	max.3
AIPASNAME	Primary passenger name. The default value is the name of the credit card holder.	optional	max.49
AIEXTRAPASNAME1	Name of extra passenger for PNRs with more than one passenger. This parameter can be repeated up to 5 times (i.e. for 5 extra passengers), changing the digit at the end of the parameter name.	optional	max.49
AICHDET	Charge details. Free text description or reference.	optional	max.49
AIAIRTAX	Airport taxes.	optional	num *100 => no decimals
AIVATAMNT	VAT amount.	optional	num *100 => no decimals
AIVATAPPL	VAT applicable flag. Supported values: D: normal VAT applicable I: no VAT on the transaction	optional	max.1
AITYPCH	Type of charge.	optional	max.2
AIEYCD	Destination area code.	optional	max.3
AIIRST	Destination area code type.	optional	max.1

The following fields can be repeated n times, changing the digit at the end of the field name.

Name	Usage		Field details
AIORCITY1	Departure airport (short).	mandatory	max.5
AIORCITYL1	Departure airport (long).	mandatory	max.20
AIDESTCITY1	Arrival airport (short).	mandatory	max.5
AIDESTCITYL1	Arrival airport (long).	mandatory	max.20
AISTOPOV1	Stopover.	optional	Possible values: the capital letters O and X. O: the passenger is allowed to stop and stay. X: the passenger is not allowed to stay.
AICARRIER1	Carrier code.	mandatory	max.4
AIBOOKIND1	Booking indicator.	optional	max.2
AIFLNUM1	Flight number.	optional	max.4
AIFLDATE1	Flight date.	optional	MM/DD/YY or YYYYMMDD
AICLASS1	Airline class.	optional	max.15

8 Appendix 3: Troubleshooting

The following section contains a non-exhaustive list of possible errors you can find in the NCERRORPLUS field:

Connection to API feature not allowed for this user

You have sent us a request with only the PSPID/password or PSPID/administrative user/password as login details. You need to create a special API user to send requests to our server. An API is a user specifically designed to be used by an application to make automatic requests to the payment platform. Please refer to the **User Manager** documentation for more information on how to create an API user.

unknown order/1/i

This error means that the IP address from which a request was sent is not an IP address the merchant had entered in the IP address field of the "Data and origin verification" tab, checks for DirectLink section of his Technical Information page. The merchant is sending us a request from a different server from the one(s) entered in the IP address field of the "Data and origin verification" tab, checks for DirectLink section.

unknown order/1/s

This error message means that the SHASign sent in your transaction request differs from the SHASign calculated at our end using the order details and the additional string (password/pass phrase) entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page.

unknown order/0/s

This error message means that the "SHASign" field in your request is empty but an additional string (password/pass phrase) has been entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, checks for DirectLink section of the Technical Information page, indicating you want to use a SHA signature with each transaction.

PSPID not found or not active

This error means the value you entered in the PSPID field does not exist in the respective environment (test or prod) or the account has not yet been activated.

no <parameter> (for instance: no PSPID)

This error means the value you sent for the obligatory <parameter> field is empty. Note: orderID is the first field we check, so if you receive the error "no orderID", it can also mean we did not receive any values at all.

<parameter> too long (for instance: currency too long)

This error means the value in your <parameter> field exceeds the maximum length.

amount too long or not numeric: ... OR Amount not a number

This error means the amount you sent in the hidden fields either exceeds the maximum length or contains invalid characters such as '.' (period) or ',' (comma) for example.

not a valid currency : ...

This error means you sent a transaction with a currency code that is incorrect or does not exist.

The currency is not accepted by the merchant

This error means you sent a transaction in a currency that has not been registered in your account details.

ERROR, PAYMENT METHOD NOT FOUND FOR: ...

This error means the PM value you sent in your hidden fields does not match any of the payment methods selected in your account, or that the payment method has not been activated in your payment methods page.

9 Appendix 4: List of Parameters to be included in SHA IN Calculation

ACCEPTURL
ADDMATCH
ADDRMATCH
AIAIRNAME
AIAIRTAX
AIBOOKIND*XX*
AICARRIER*XX*
AICHDET
AICLASS*XX*
AICONJTI
AIDESTCITY*XX*
AIDESTCITYL*XX*
AIEXTRAPASNAME*XX*
AIEYCD
AIFLDATE*XX*
AIFLNUM*XX*
AIIRST
AIORCITY*XX*
AIORCITYL*XX*
AIPASNAME
AISTOPOV*XX*
AITIDATE
AITINUM
AITYPCH
AIVATAMNT
AIVATAPPL
ALIAS
ALIASOPERATION
ALIASUSAGE
ALLOWCORRECTION
AMOUNT
AMOUNT*XX*
AMOUNTHTVA
AMOUNTTVA
BACKURL
BGCOLOR
BRAND
BRANDVISUAL
BUTTONBGCOLOR
BUTTONTXTCOLOR
CANCELURL
CARDNO
CATALOGURL
CAVV_3D
CAVVALGORITHM_3D
CERTID
CHECK_AAV
CIVILITY
CN
COM
COMPLUS
COSTCENTER
COSTCODE
CREDITCODE
CUID
CURRENCY
CVC
DATA
DATATYPE

DATEIN
DATEOUT
DECLINEURL
DISCOUNTRATE
ECI
ECOM_BILLTO_POSTAL_CITY
ECOM_BILLTO_POSTAL_COUNTRYCODE
ECOM_BILLTO_POSTAL_NAME_FIRST
ECOM_BILLTO_POSTAL_NAME_LAST
ECOM_BILLTO_POSTAL_POSTALCODE
ECOM_BILLTO_POSTAL_STREET_LINE1
ECOM_BILLTO_POSTAL_STREET_LINE2
ECOM_BILLTO_POSTAL_STREET_NUMBER
ECOM_CONSUMERID
ECOM_CONSUMERORDERID
ECOM_CONSUMERUSERALIAS
ECOM_PAYMENT_CARD_EXPDATE_MONTH
ECOM_PAYMENT_CARD_EXPDATE_YEAR
ECOM_PAYMENT_CARD_NAME
ECOM_PAYMENT_CARD_VERIFICATION
ECOM_SHIPTO_COMPANY
ECOM_SHIPTO_DOB
ECOM_SHIPTO_ONLINE_EMAIL
ECOM_SHIPTO_POSTAL_CITY
ECOM_SHIPTO_POSTAL_COUNTRYCODE
ECOM_SHIPTO_POSTAL_NAME_FIRST
ECOM_SHIPTO_POSTAL_NAME_LAST
ECOM_SHIPTO_POSTAL_POSTALCODE
ECOM_SHIPTO_POSTAL_STREET_LINE1
ECOM_SHIPTO_POSTAL_STREET_LINE2
ECOM_SHIPTO_POSTAL_STREET_NUMBER
ECOM_SHIPTO_TELECOM_FAX_NUMBER
ECOM_SHIPTO_TELECOM_PHONE_NUMBER
ECOM_SHIPTO_TVA
ED
EMAIL
EXCEPTIONURL
EXCLPMLIST
EXECUTIONDATE*XX*
FIRSTCALL
FLAG3D
FONTTYPE
FORCECODE1
FORCECODE2
FORCECODEHASH
FORCEPROCESS
FORCETP
GENERIC_BL
GIROPAY_ACCOUNT_NUMBER
GIROPAY_BLZ
GIROPAY_OWNER_NAME
GLOBORDERID
GUID
HDFONTTYPE
HDTBLBGCOLOR
HDTBLTXTCOLOR
HEIGHTFRAME
HOMEURL
HTTP_ACCEPT
HTTP_USER_AGENT
INCLUDE_BIN
INCLUDE_COUNTRIES
INVDATE
INVDISCOUNT
INVLEVEL
INVORDERID
ISSUERID

ITEMCATEGORY*XX*
ITEMDISCOUNT*XX*
ITEMID*XX*
ITEMNAME*XX*
ITEMPRICE*XX*
ITEMQUANT*XX*
ITEMUNITOFMEASURE*XX*
ITEMVATCODE*XX*
LANGUAGE
LEVEL1AUTHCPC
LIDEXCL*XX*
LIMITCLIENTSCRIPTUSAGE
LINE_REF
LIST_BIN
LIST_COUNTRIES
LOGO
MERCHANTID
MODE
MTIME
MVER
NETAMOUNT
OPERATION
ORDERID
ORIG
OR_INVORDERID
OR_ORDERID
OWNERADDRESS
OWNERADDRESS2
OWNERCTY
OWNERTELNO
OWNERTOWN
OWNERZIP
PAIDAMOUNT
PARAMPLUS
PARAMVAR
PAYID
PAYMETHOD
PM
PMLIST
PMLISTPMLISTTYPE
PMLISTTYPE
PMLISTTYPEPMLIST
PMTYPE
POPUP
POST
PSPID
PSWD
REF
REFER
REFID
REFKIND
REF_CUSTOMERID
REF_CUSTOMERREF
REMOTE_ADDR
REQGENFIELDS
RTIMEOUT
RTIMEOUTREQUESTEDTIMEOUT
SCORINGCLIENT
SETT_BATCH
SID
STATUS_3D
SUBSCRIPTION_ID
SUB_AM
SUB_AMOUNT
SUB_COM
SUB_COMMENT
SUB_CUR

SUB_ENDDATE
SUB_ORDERID
SUB_PERIOD_MOMENT
SUB_PERIOD_MOMENT_M
SUB_PERIOD_MOMENT_WW
SUB_PERIOD_NUMBER
SUB_PERIOD_NUMBER_D
SUB_PERIOD_NUMBER_M
SUB_PERIOD_NUMBER_WW
SUB_PERIOD_UNIT
SUB_STARTDATE
SUB_STATUS
TAAL
TAXINCLUDED*XX*
TBLBGCOLOR
TBLTXTCOLOR
TID
TITLE
TOTALAMOUNT
TP
TRACK2
TXTBADDR2
TXTCOLOR
TXTOKEN
TXTOKENTXTOKENPAYPAL
TYPE_COUNTRY
UCAF_AUTHENTICATION_DATA
UCAF_PAYMENT_CARD_CVC2
UCAF_PAYMENT_CARD_EXPDATE_MONTH
UCAF_PAYMENT_CARD_EXPDATE_YEAR
UCAF_PAYMENT_CARD_NUMBER
USERID
USERTYPE
VERSION
WBTU_MSISDN
WBTU_ORDERID
WEIGHTUNIT
WIN3DS
WITHROOT