

Integrate with ePDQ DirectLink (server-to-server)

Table of contents

1. Introduction

2. General procedures and security settings

2.1 API user

2.2 Request form

2.3 Security

2.3.1 Encryption

2.3.2 IP address

2.3.3 SHA signature

2.4 Response parsing

3. Request a new order

3.1 Request URL

3.2 Request parameters

3.3 Test page

3.4 Excluding specific payment methods

3.5 Order request using 3-D Secure

3.6 Split credit/debit cards

4. Order response

4.1 Duplicate request

5. Direct Maintenance

5.1 Maintenance request

5.1.1 Request URL

5.1.2 Request parameters

5.1.3 Test page

5.2 Maintenance response

5.3 Duplicate request

6. Direct Query

6.1 Query request

Integrate with ePDQ DirectLink (server-to-server)

6.1.1 Request URL

6.1.2 Request parameters

6.1.3 Test page

6.2 Query response

6.2.1 Transactions processed with e-Commerce (hosted payment page)

6.3 Possible response statuses

6.4 Direct Query as fallback

7. Data Controller privacy policy request

7.1 Query request

7.1.1 Request URL

7.1.2 Request-parameters

7.1.3 Test-page

7.2 Query response

8. Optional: Visa Additional Authorisation Data

1. Introduction

ePDQ DirectLink allows you to set up a server-to-server integration with our platform. The customer remains on a page of your own that will securely send the payment data to our servers.

You can also use DirectLink for [maintenance of transactions](#), whether they were initiated in DirectLink or in e.g. e-Commerce mode.

Using DirectLink, there is no contact between our system and the merchant's (your) customer. Your system 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 your server in XML format. Your programme reads the response and resumes its processing.

You are therefore responsible for collecting and storing your customer's confidential payment details and must guarantee the confidentiality and security of these details by means of encrypted web communication and server security.

In order to store personal and card data, you need to be PCI compliant.

2. General procedures and security settings

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

2.1 API user

An API (Application Program Interface) user is needed to make DirectLink requests.

In general it's a user specifically designed to be used by an application to make automatic requests to the payment platform.

You can create an API user in your ePDQ account via "Configuration" > "Users". Select "New user" and fill the required fields.

To make the new user an API user, make sure to enable the "Special user for API (no access to admin.)" box.

User's Data

UserID JM-API-User *

REFID gvetest

User type PSPID

User's name John Mills *

E-mail address johnmills@jmindustries.com *

Timezone (GMT+01:00) Brussels, Copenhagen, Madri... ▼

Automatically adjust to daylight saving changes

User created by gvetest/gvetest/PSPID

Profile Admin ▼

Scope limited to user?

Special user for API (no access to admin.) [Related FAQ](#)

Access rights Fraud detection
 Technical information
 Payment methods

To confirm the modification, please enter your own password *

Even though various user profiles are available for an API user, we strongly recommend you to configure this user with the "Admin" profile. If you want to limit the rights for maintenance of transactions (refunds, cancellations etc.), you can still change the user profile to e.g. "Encoder".

If you are not sure, we recommend you to choose the "Admin" profile, otherwise go to [User profiles](#) (User Manager) for more information.

The password for an API user does not have to be changed regularly. This is more convenient when the password has to be hard-coded into your application. However, we recommend you to change the password from time to time.

For more information about User types and how to change the API user's password, go to [User types](#) (User Manager).

2.2 Request form

For new order requests, maintenance requests and direct queries, you must send requests with certain parameters to specific URLs. The new order/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".

Integrate with ePDQ DirectLink (server-to-server)

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.

2.3 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.

2.3.1 Encryption

DirectLink is built on a robust, 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 TLS 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 TLS certificate.

When we receive a request, we check the level of encryption. We allow merchants to connect to us only in secure https mode using TLS protocols and we strongly recommend to use the most recent and secure versions which are currently TLS 1.1 and 1.2.

2.3.2 IP address

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

If the originating IP address has not been declared in the given IP address field, 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.

2.3.3 SHA signature

The SHA signature is based on the principle of your (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 your order request. Our system reconstructs this signature to check the integrity of the order data sent to us in the request.

Go to [SHA-IN Signature](#) (ePDQ e-Commerce documentation) - the principle is the same in e-Commerce and DirectLink mode.

For DirectLink, the SHA-IN passphrase needs to be configured in the "Checks for DirectLink" section of the "Data and origin verification" tab in your Technical information page.

2.4 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 Request URL

- The request URL in the TEST environment is <https://mdepayments.epdq.co.uk/ncol/test/orderdirect.asp>.
- The request URL in the PRODUCTION environment is <https://payments.epdq.co.uk/ncol/prod/orderdirect.asp>.

Change "test" to "prod"

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 processed by the acquirers/banks.

3.2 Request parameters

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

Format: AN= Alphanumeric / N=Numeric, maximum allowed amount of characters

Field	Description	Format	Mandatory
PSPID	Your affiliation name in our system.	AN, 30	Yes
ORDERID	Your unique order number (merchant reference).	AN, 40	Yes
USERID	Name of your application (API) user. Please refer to the User Manager documentation for information on how to create an API user.	AN, 20 (min 2)	Yes
PSWD	Password of the API user (USERID).	AN	Yes
AMOUNT	Amount to be paid, MULTIPLIED BY 100 as the format of the amount must not contain any decimals or other separators.	N, 15	Yes
CURRENCY	ISO alpha order currency code, for example: EUR, USD, GBP, CHF, etc.	AN, 3	Yes
CARDNO	Card/account number.	AN, 21	Yes
ED	Expiry date.	MM/YY or MMY	Yes
COM	Order description.	AN, 100	No
CN	Customer name.	AN, 35	No
EMAIL	Customer's email address.	AN, 50	No
SHASIGN	Signature (hashed string) to authenticate the data (see SHA-IN Signature).	AN, 128	Yes
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.	N, 5	Yes
ECOM_PAYMENT_CARD_VERIFICATION	Alternative to CVC: date of birth / issue number / etc. (depending on country/bank)	N, 5	No
OWNERADDRESS	Customer's street name and number.	AN, 50	No
OWNERZIP	Customer's postcode.	AN, 10	No
OWNERTOWN	Customer's town/city name.	AN, 40	No
OWNERCTY	Customer's country, e.g. BE, NL, FR, etc.	AN, 2	No
OWNERTELNO	Customer's telephone number.	AN, 30	No
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	A, 3	Yes

Integrate with ePDQ DirectLink (server-to-server)

Field	Description	Format	Mandatory
	<p>request, this will overwrite the default value.</p> <p>Possible values:</p> <ul style="list-style-type: none"> 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). <p>Optional:</p> <ul style="list-style-type: none"> PAU: Request for pre-authorization: In agreement with your acquirer you can use this operation code to temporarily reserve funds on a customer's card. This is a common practice in the travel and rental industry. PAU/pre-authorization can currently only be used on MasterCard and Visa transactions and is supported by selected acquirers. This operation code cannot be set as the default in your ePDQ account. Should you use PAU on transactions via acquirers or with card brands that don't support pre-authorization, these transactions will not be blocked but processed as normal (RES) authorizations. 		
WITHROOT	Adds a root element to our XML response. Possible values: 'Y' or empty.	Y or <empty>	No
REMOTE_ADDR	Customer's IP address (for Fraud Detection Module only). If a country check does not need to be performed on the IP address, send 'NONE'.	AN	No
RTIMEOUT	<p>Request timeout for the transaction (in seconds, value between 30 and 90)</p> <p>Important: The value you set here must be smaller than the time out value in your system (!)</p>	N, 2	No
ECI	<p>Electronic Commerce Indicator.</p> <p>You can configure a default ECI value in your account's Technical information page, "Global transaction parameters" tab, "Default ECI value" section. When you send an ECI value in the request, this will override the default ECI value.</p> <p>Possible (numeric) values:</p> <ul style="list-style-type: none"> 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) 	N, 2	No

> If your business falls under the Merchant Category Code (MCC) 6012, please find extra info and parameters [here](#).

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

3.3 Test page

Our test page to send order requests in DirectLink can be found here: <https://mdepayments.epdq.co.uk/ncol/test/testodl.asp>.

3.4 Excluding specific payment methods

If there are payment methods you don't want a customer to be able to pay with, you can use a parameter to do so. This is particularly useful for sub-brands, when you want to accept a brand (e.g. MasterCard) but not one of its sub-brands (e.g. Maestro).

The parameter is the following:

Field	Usage
EXCLPMLIST	List of payment methods and/or credit card brands that should NOT be used. Values must be separated by a ";" (semicolon).

If a customer tries paying with a card linked to a payment method and/or (sub)brand thT you've excluded BY using the EXCLPMLIST parameter, the error message "Card number incorrect or incompatible" will be returned with the NCERRORPLUS return field.

3.5 Order request using 3-D Secure

Our system supports the usage of [3-D Secure with DirectLink](#).

Important

- If you wish to use 3-D Secure with DirectLink, you need to have the D3D option activated in your account.
- Some acquiring banks require the use of 3-D Secure. Please check with your acquirer if this is the case for you.

3.6 Split credit/debit cards

The functionality to split VISA and MasterCard into a debit and a credit payment method allows you to offer them to your customers as two different payment methods (e.g. VISA Debit and VISA Credit), or you can decide only to accept one of both split brands.

To use the split of credit and debit cards via DirectLink, you need to include the CREDITDEBIT parameter in the fields that you send to the orderdirect.asp page (and therefore also include in the SHA-IN calculation!).

Field	Format
CREDITDEBIT	"C": credit card "D": debit card

Related error: When the buyer selects the debit card method but next enters a credit card number, an error code will be returned: 'Wrong brand/Payment method was chosen'.

If the payment is successfully processed with the CREDITDEBIT parameter, the same parameter will also be returned in the XML response, and/or can be requested with a Direct Query. However, whereas the submitted values are C or D, the return values are "CREDIT" or "DEBIT".

You will also find these return values in transaction overview via "View transactions" and "Financial history", and in reports you may download afterwards.

Configuration in your account

The "split" functionality can also be activated and configured per payment method, in your ePDQ account. Go to [Split Credit/Debit Cards](#) for more information.

4. Order response

Our server returns an XML response to a 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 ncreponse tag attributes:

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

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

4.1 Duplicate request

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

5. Direct Maintenance

A direct maintenance request from your application allows you to:

- Perform a data capture (payment) of an authorised order automatically (as opposed to manually in the back office);
- Cancel an authorisation of an order;
- Renew an authorisation of an order;
- Refund a paid order.

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

5.1 Maintenance request

5.1.1 Request URL

- The request URL in the TEST environment is <https://mdepayments.epdq.co.uk/ncol/test/maintenancedirect.asp>.
- The request URL in the PRODUCTION environment is <https://payments.epdq.co.uk/ncol/prod/maintenancedirect.asp>.

Change "test" to "prod"

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 working with real orders, your maintenance transactions will be sent to the test environment and will not be sent to the acquirers/banks.

5.1.2 Request parameters

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

Field	Description
AMOUNT	<p>Order amount multiplied by 100.</p> <p>This is only required when the amount of the maintenance differs from the amount of the original authorisation. However, we recommend its use in all cases.</p> <p>Our system will check that the maintenance transaction amount is not higher than the authorisation/payment amount.</p>
OPERATION	<p>Possible values:</p> <ul style="list-style-type: none"> • REN: renewal of authorisation, if the original authorisation is no longer valid. • DEL: delete authorisation, leaving the transaction open for further potential maintenance operations. • DES: delete authorisation, closing the transaction after this operation. • SAL: partial data capture (payment), leaving the transaction open for another potential 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 another potential refund. • RFS: (last) partial or full refund (on a paid order), closing the transaction after this refund. <p>Please note that with DEL and DES that not all acquirers support the deletion of an authorisation. If your acquirer does not support DEL/DES, we will nevertheless simulate the deletion of the authorisation in the back office.</p>
ORDERID	You can send the PAYID or the orderID to identify the original order. We recommend the use of the PAYID.
PAYID	
PSPID	Your account's PSPID
PSWD	Password of your API-user
SHASIGN	Signature (hashed string) to authenticate the data (see SHA-IN-signature)
USERID	Your API-user

5.1.3 Test page

You can test direct maintenance requests here: <https://mdepayments.epdq.co.uk/ncol/test/testdm.asp>

5.2 Maintenance response

Integrate with ePDQ DirectLink (server-to-server)

Our server returns an XML response to the maintenance request:

Example of an XML response to a direct maintenance request

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

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

Field	Description
ACCEPTANCE	Acceptance code returned by acquirer
AMOUNT	Order amount (not multiplied by 100)
CURRENCY	Order currency
NCERROR	Error code
NCERRORPLUS	Explanation of the error code
NCSTATUS	First digit of NCERROR
ORDERID	Your order reference
PAYID	Payment reference in our system
PAYIDSUB	The history level ID of the maintenance operation on the PAYID
STATUS	Transaction status (Possible statuses)

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

5.3 Duplicate request

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

6. Direct Query

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 you 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 (see [Consult your transactions and Batch](#)).

6.1 Query request

6.1.1 Request URL

- The request URL in the TEST environment is <https://mdepayments.epdq.co.uk/ncol/test/querydirect.asp>
- The request URL in the PRODUCTION environment is <https://payments.epdq.co.uk/ncol/prod/querydirect.asp>

Change "test" to "prod"

Replace "test" with "prod" in the request URL when you switch to your production account.

6.1.2 Request parameters

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

Field	Description
ORDERID	You can send the PAYID or the ORDERID to identify the original order. We recommend the use of the PAYID.
PAYID	
PAYIDSUB	You can indicate the history level ID if you use the PAYID to identify the original order (optional).
PSPID	Your account's PSPID
PSWD	Password of your API-user
USERID	Your API-user

6.1.3 Test page

You can test direct query requests here: <https://mdepayments.epdq.co.uk/ncol/test/testdq.asp>.

6.2 Query response

Our server returns an XML response to the request:

Example of an XML response to a direct query

```
<?xml version="1.0"?>
<nresponse orderID="99999" PAYID="111111" 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 nresponse tag attributes:

Field	Usage
ACCEPTANCE	Acceptance code returned by acquirer
amount	Order amount (<u>not</u> multiplied by 100)
BRAND	Card brand or similar information for other payment methods
CARDNO	The masked credit card number
currency	Order currency

Integrate with ePDQ DirectLink (server-to-server)

Field	Usage
ECI	Electronic Commerce Indicator
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
NCERROR	Error code
NCERRORPLUS	Explanation of the error code
NCSTATUS	First digit of NCERROR
orderID	Your order reference
PAYID	Payment reference in our system
PAYIDSUB	The history level ID of the maintenance operation on the PAYID
PM	Payment method
STATUS	Transaction status

The standard nresponse 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) in their accounts. Please refer to the respective option documentation for more information on extra response attributes linked to the option.

6.2.1 Transactions processed with e-Commerce (hosted payment page)

If the transaction whose status you want to check was processed with e-Commerce (hosted payment page), you may also receive the following additional attributes (providing you sent these fields with the original e-Commerce transaction).

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

*Please check the [Variable feedback parameters](#) (e-Commerce documentation).

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

```
<nresponse 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"/>
```

6.3 Possible response statuses

The STATUS field will contain the status of the transaction (see [Possible statuses](#)).

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

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

6.4 Direct Query as fallback

The response times for a DirectLink transaction request are generally a few seconds; however, some acquirers may have longer response times.

If you haven't received a response from our system after 30 seconds, you can send a request to querydirect.asp, asking for the status of your most recent transaction sent to orderdirect.asp. If you receive an immediate reply containing a non-final status for the transaction, there might be issues on the acquirer's end.

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

Integrate with ePDQ DirectLink (server-to-server)

Note

- 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.
- 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.
- Please use these checks judiciously to avoid bombarding our servers with requests, otherwise we might have to restrict your access to the querydirect.asp page.

Important

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

7. Data Controller privacy policy request

Based on GDPR article 12, 13 & 14, a Data Controller has the obligation to inform its end-customers about the future processing of their personal data. Such information should be made specific based on the type of personal data to be filled-in for a specific transaction (e.g.: selected payment method, controller/processor, acquirer, fraud). The result should be available and visible at the moment of the data collection and the cardholder should be offered with a printable and downloadable version of it.

NOTE:

Barclaycard / Smartpay i act as Data Controller for our Fraud Expert and Alternative Payments products and services.

Per the GDPR policy, you need to display the information to your customer before they validate their transaction. This information should ideally be displayed on the same page as where your customer fills in their card/account credentials.

The below privacy policy request allows you to retrieve all the information you need to display to your customer about our services in order to be compliant with the GDPR regulation.

7.1 Query request

7.1.1 Request URL

- The request URL in the TEST environment is <https://mdepayments.epdq.co.uk/ncol/test/privacy-policy.asp>
- The request URL in the PRODUCTION environment is <https://payments.epdq.co.uk/ncol/prod/privacy-policy.asp>
Change "test" to "prod"

Replace "test" with "prod" in the request URL when you switch to your production account.

7.1.2 Request-parameters

The following table contains the mandatory request parameters to be sent to your customer regarding the usage of their privacy information:

Field	Format	Description
USERID	String	Your API-user
PSWD	String	Your API-user password
PSPID	String	Your account's PSPID
BRAND	String (e.g. Visa)	Optional: Payment method brand You can send this field multiple times to get the result of several brands at once. <ul style="list-style-type: none"> • Sending no brand is the same as sending all your active brands. • Empty/wrong formatted brands are ignored.
LANGUAGE	ISO 639-1: Two-letter codes (e.g. FR)	Optional: The language in which you want to retrieve the text. If not provided, the text will be returned into the merchant configured language.

7.1.3 Test-page

You can test direct query requests here: <https://mdepayments.epdq.co.uk/ncol/test/privacy-policy.asp>

7.2 Query response

The following is a list of XML elements and the returned XML responses examples for different outcomes.

Name	Format	Description
Response	Complex	Root node, always present
Response.Status	String, possible values : Success, SuccessWithWarnings, Error	Always present
Response.Body	Complex	Present only when Response.Status = Success or SuccessWithWarnings
Response.Body.Html	String / html	Empty if Response.Status = SuccessWithWarnings & Response.Warnings.Warning.Code = NoContent
Response.Errors	Complex	Present only when Response.Status = Error

Integrate with ePDQ DirectLink (server-to-server)

Response.Errors.Error	Complex	Can occur multiple times inside an <Errors> node
Response.Warnings	Complex	Present only when Response.Status = SuccessWithWarnings or Error
Response.Warnings.Warning	Complex	Occurs multiple times inside a <Warnings> node
Response.Errors.Error.Code Response.Warnings.Warning.Code	String, possible values : •Inside an <Error> node : Unauthorized, InternalServerError •Inside a <Warning> node : NoContent	Always present in an <Error> or <Warning> node
Response.Errors.Error.Message Response.Warnings.Warning.Message	String	Optional

If you face Response.Status=Error, please refer to the Response.Errors.Error to fix it.

The following are two successful examples:

1. Example of an XML response for success with warnings. This example displays if no privacy information needs to be disclosed to the customer.

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Status>SuccessWithWarnings</Status>
  <Warnings>
    <Warning>
      <Code>NoContent</Code>
    </Warning>
  </Warnings>
  <Body>
    <Html/>
  </Body>
</Response>
```

2. Example of an XML response for success with content. The example shows a 2 section display.

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Status>Success</Status>
  <Body>
    <Html><![CDATA[<ul><li><h2>Title 1</h2><p>Content 1</p></li><li><h2>Title 2 (VISA, American Express)</h2><p>Content 2</p></li></ul>]]></Html>
  </Body>
</Response>
```

8. Optional: Visa Additional Authorisation Data

For ePDQ Essential, ePDQ Extra & ePDQ Extra Plus

In order to reduce fraud, Visa has introduced additional transaction authorisation fields for any UK merchant defined as a Financial Institution. These fields must be captured during your order preparation and submitted to ePDQ, regardless of whether you have integrated via the Hosted Payment Page e-Commerce) or DirectLink.

Current fraud detection tools may not give card issuing banks sufficient information to validate transactions in this business sector. With this additional data, issuers are able to make a more informed decision.

To comply with these requirements you need to submit the following additional fields in the authorisation requests you send to ePDQ. If you use the SHA-IN passphrase, these fields will need to be appropriately included in the [SHA-IN calculation](#).

Field	Description	Format	Example
RECIPIENTACCOUNTNUMBER	Recipient's account number OR partially masked credit card number	AlphaNum / 10 char. If the number is longer than 10 characters, we only pass the 6 first digits + the last 4 digits.	"12345ABCDZ6789" -> "12345A6789"
RECIPIENTDOB	Recipient's date of birth	DD/MM/YYYY / Num. / 10 char. Division slashes must be entered. Our system will convert the date as follows: YYYYMMDD	"02/03/1982" -> "19820302"
RECIPIENTLASTNAME	Recipient's surname	Alpha / 6 char. If the value is longer than 6 characters, we only pass the 6 first characters.	"Day-O'Reilly" -> "DAYORE"
RECIPIENTZIP	Recipient's postcode	AlphaNum / 6 char. You must only enter the first part of the postcode, up to the space (e.g. 3 or 4 digits).	"MK4 " -> "MK4" "MK46 " -> "MK46"

More information about these fields can be found in your ePDQ account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.

Note: If special characters, other than a space, ' (apostrophe), * (asterisk), \$ (dollar sign), / (forward slash) or - (dash) are inserted in a field, the whole field content will be emptied automatically when submitted.

When setting up your account, Barclaycard will ensure that it is enabled to support these additional fields. If the fields are not available to you and you believe that this Visa requirement applies to your business, then please contact us at epdqsupport@barclaycard.co.uk.

If enabled, the following flag should be visible in the Visa configuration page of your ePDQ account:

MCC 6012 retailer enabled