iDEAL QR - Merchant Implementation Guidelines (EN)

Version: 1.5
Date: 19-02-2018

Versions

...

version	Adjustments
1.0	First version for extended pilot
1.1	- max_amount and min amount introduced - general calrifictions
1.2	- Change in fields amount_changeable and one_off
1.3	Optionality merchant_sub-id and amount removed
1.4	Adjusted signature key procedure (for calls/responses from back-end only)
1.5	- HMAC signature process further clarified - Error response and error codes added for clarification

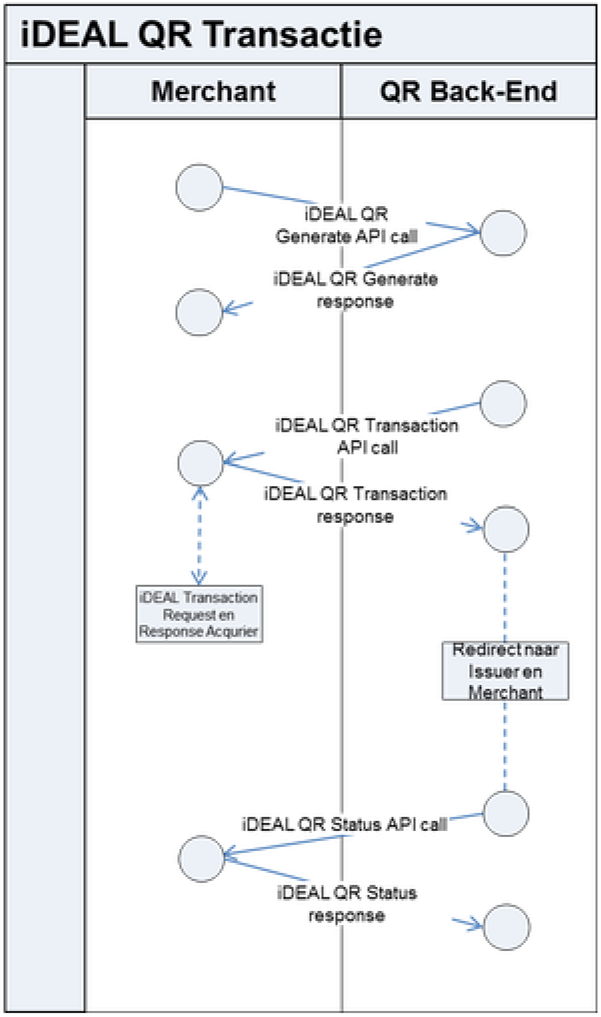
Contents

1. What is iDEAL QR?

In addition to the online domain, iDEAL can also be used in the 'physical context' using iDEAL QR. Merchants can generate an iDEAL QR code that allows consumers to initiate an iDEAL payment using the iDEAL App. The QR code is used only to initiate a regular mobile iDEAL transaction, which means that it does not carry out the iDEAL transaction itself. This means that the iDEAL 'four-corner message' model remains intact; only external components are added.

As with the existing iDEAL protocols between the Merchant and the Acquirer, iDEAL QR has three dedicated protocols between the Merchant and a central iDEAL QR back-end. These protocols are: iDEAL QR Generate, iDEAL QR Transaction and iDEAL QR Status. Merchants can use the iDEAL QR Generate API protocol to create an iDEAL QR code. According to the iDEAL QR Transaction API protocol, after the QR code has been scanned, an iDEAL TransactionRequest is generated from the QR back-end to the Merchant and the resulting IssuerAuthenticationURL is sent back to the QR back-end and the iDEAL app by the Merchant. With the iDEAL QR Status API protocol, the QR back-end requests the status of the iDEAL transaction from the Merchant.

The chart below shows the protocols; the dotted lines represent the regular iDEAL protocol.


iDEAL QR Transactie
Merchant
QR Back-End
iDEAL QR Generate API call
iDEAL QR Generate response
iDEAL QR Transaction API call
iDEAL QR Transaction response
iDEAL Transaction Request en Response Acqurier
Redirect naar Issuer en Merchant
iDEAL QR Status API call
iDEAL QR Status response

Schematic representation of the steps in Ideal QR transactions

2. Registration and Onboarding

In order to be able to use iDEAL QR, the Merchant must have an iDEAL contract with one of the Acquiring Parties affiliated with iDEAL. The Merchant should contact its Acquirer in order to register for iDEAL QR to be able to connect with the central iDEAL QR back-end. After registration has been accepted, the Merchant can have iDEAL QR codes generated and connect to the QR back-end in order to submit iDEAL payment requests from scanned QR codes.

The following information must be registered prior to participating in the QR protocol.

Data element	Description
`MerchantID`	MerchantID of Merchant, as known to the Acquirer.
`LegalName`	The legal name of the Merchant, as registered with the Acquirer. This is only used for registration purposes, not as the name of the beneficiary that will be visible on iDEAL QR screens. The QR Beneficiary Name field is used for this (see below under 'QR Generate').
`Merchant API URLs`	The URLs that the Merchant uses to communicate with the QR back-end. This may be one URL for both API's (initiate and status) or for each API a separate URL
`Contact name`	Name of the person who can be contacted in the event of operational problems.
`Contact email`	Email address of the contact person.
`Contact phone`	Telephone number of the contact person.

After registration, the Merchant will receive:

Data element	Description
Merchant Token	A secret identification number specific to the Merchant which the iDEAL QR back-end uses to identify and authenticate the Merchant. The Merchant can use this token to have iDEAL QR codes generated by the iDEAL QR back-end.
API URLs QR backend	The API URLs of the Merchant to which the QR back-end sends its requests.
Secret signing key	A secret symmetric key with which the iDEAL QR back-end signs all their messages and with which the Merchant can authenticate the incoming messages. See paragraph 8 on security.

Note
The specified Merchant API URL used for the iDEAL QR Transaction protocol must be available at all times to enable Consumers to successfully scan the iDEAL QR. If the URL is not available, Consumers cannot initiate iDEAL transactions by scanning QR codes.

3. Message format

The specifications for the iDEAL QR protocol message format for messages between the Merchant and the QR back-end are the same as for the standard iDEAL protocol message format. However, there are a number of differences:

- For the body of the HTTP POST and 200 OK response message, the JSON message format is used instead of XML.
- The content type of the HTTP header refers to the JSON message format.
- In messages from the iDEAL QR back-end a signature is provided in the HTTP header under *x-ideal-qr-hash*. See paragraph 8 for further information on security.

Example (iDEAL QR Generate call):

```
POST /ideal-qr/v1.0/generate HTTP/1.1
Content-type: application/json, charset=UTF-8
{
"merchant_token": "784aea4c-e36c-4a4b-b164-f9818aaeaf5c",
"merchant_sub_id": 5,
"amount": 24.95,
"amount_changeable": false,
"description": "Product Y",
"one_off": false,
"expiration": "2016-05-14 00:00",
"beneficiary": "Organisatie X",
"purchase_id": "PO1234567",
"size": 1000
}
```

4. Generate iDEAL QR code: IDEAL QR Generate protocol

Once the Merchant has registered and the registration process is complete, it can generate iDEAL QR codes from the iDEAL QR back-end using the Generate API protocol. In this protocol, the relevant details associated with the specific QR code (such as the amount and description) will be stored in the iDEAL QR back-end and coupled with a unique iDEAL QR ID. The QR code generated only contains this iDEAL QR ID (packaged in a URL), therefore no payment information is included in the QR code.

4.1 iDEAL QR Generate API call

To generate an iDEAL QR code, the Merchant must perform a Generate API call. This consists of a JSON message that is sent to the QR back-end as the body of an HTTP POST request. The table below shows the fields in the Generate API call and the required format.

Name	Description	Format
`merchant_token`	This is the ID that the Merchant receives for onboarding at the QR back-end. It identifies and authenticates the Merchant at the QR back-end.	AN..36
`amount`	The pre-entered amount for the specific QR code. A full stop (.) is used to show the decimal point. Amount cannot be 0 (as iDEAL transactions of 0 are not possible)	DEC (12.2)
`amount_changeable`	This field indicates whether the Consumer can edit the amount in the app. Enter 'false' when the amount must not be fixed and 'true' when the amount must be editable by the user. Please note that for fixed amounts, the amount must be entered and it cannot be 0.	true/false
`amount_max` (conditional)	In this field a maximum amount can be provided if the amount is not fixed. It must be larger than 0 and larger than "amount".	DEC (12.2)
`amount_min` (optional)	In this field a mimimum amount can be provided if the amount is not fixed. It must be larger than 0 and smaller than "amount".	DEC (12.2)
`description`	The payment description shown in the app and specified in the iDEAL messages protocol.	AN..max 35
`one_off`	This field indicates whether the iDEAL QR code may be used only once or whether it can be used multiple times.	true/false
`expiration`	The date and time at which the QR code expires and can no longer be used for iDEAL transactions.	DT yyyy-mm-dd hh:mm
`beneficiary`	The name of the recipient of the iDEAL QR transaction displayed in the app.	AN..100
`merchant_sub_id`	The Merchant subID as also used for the iDEAL transaction. Use 0 when no sub-id is applicable.	N..Max 6
`purchase_id`	The reference also provided by the Merchant in the iDEAL protocol.	ANS..Max 35
`size`	In this field, the Merchant indicates the size of the QR code in pixels. The value must be between 100 and 2000 px.	N..100-2000

Fields in the iDEAL QR Generate API call

4.2 iDEAL QR Generate Response

In response to the Generate API call, the Merchant receives a Generate Response in an HTTP 200 OK message with the JSON data elements in the body. This message contains the reference to the generated QR codes and a unique QR code ID.

The table below shows all the fields in the Generate Response and the required format.

Name	Description	Format
`qr_url`	The URL at which the QR code .png file is located in the specified format and where this is available for collection by the Merchant.	AN
`qr_id`	A unique random code used to identify the QR code and the associated payment details.	AN..36

Fields in the iDEAL QR Generate response

The Merchant can then use the provided IDEAL QR code for distribution, but it cannot be modified. Additionally, for correct display of the QR code, the guidelines specified by the Merchant's Acquirer must also be followed as stated in the iDEAL QR Style Guide.

5. Transaction request after scanning the iDEAL QR code: iDEAL QR Transaction Protocol

After the Consumer has scanned the iDEAL QR code with the iDEAL App, the transaction data will be displayed on the screen. The Consumer then has the option of changing the amount and/or choosing a preferred bank, after which the iDEAL QR transaction is confirmed. When the transaction is confirmed, the iDEAL QR back-end sends an iDEAL QR Transaction API call to the Merchant. This serves as a trigger to initiate the iDEAL transaction protocol from the Merchant to the Acquirer. The Merchant must send the IssuerauthenticationURL that it receives from the Issuer (via the Acquirer) as part of the iDEAL transaction protocol back to the iDEAL QR backend as the API response.

5.1 iDEAL QR Transaction API call

When the Consumer confirms the iDEAL QR transaction in the iDEAL app, the QR back-end sends an iDEAL QR Transaction API call to the Merchant. This consists of a JSON message that the Merchant sends to the earlier specified API URL by means of an HTTP POST request.

The table below shows all the fields in the iDEAL QR Transaction API call and the required format.

Name	Description	Format
`merchant_id`	The MerchantID provided when the Merchant registers, as stored at the QR back-end.	PN..9
`qr_id`	A unique random code used to identify the QR code and the associated payment details.	AN..36
`issuer_id`	The iDEAL IssuerID that corresponds to the preferred bank specified by the Consumer in the iDEAL app for the payment of this specific iDEAL QR transaction.	ANS..Max 11
`amount`	The amount confirmed by the consumer in the iDEAL App after scanning the QR code. Only this amount must be included (in €) in the iDEAL transaction request. A full stop (.) is used for the decimal point.	DEC (12.2)
`purchase_id`	This is the PurchaseID issued by the Merchant when the QR code is generated, and also displayed to the Consumer as a reference in the iDEAL App after the QR code has been scanned. Once issued, this PurchaseID must also be used in the iDEAL Transaction Request.	ANS..Max 35
`merchant_sub_id`	The Merchant subID as also used for the iDEAL transaction. Use 0 when no sub-id is applicable.	N..Max 6
`description`	The description provided when the QR code was generated and that the Consumer has received and confirmed.	AN..Max 35

Fields in the iDEAL QR Transaction API call

When the Merchant receives the iDEAL QR Transaction API call, the Merchant should automatically initiate an iDEAL TransactionRequest (message B) to the Acquirer. This request must include the fields from the iDEAL QR Transaction API call. The other required fields for the TransactionRequest, such as the MerchantReturnURL, must be added by the Merchant.

5.2 iDEAL QR TransactionResponse

After sending the iDEAL TransactionRequest described above (message B), the Merchant receives a TransactionResponse message back from the Acquirer (B'). The Merchant must be able to recognize that this response relates to an iDEAL QR transaction and automatically send a response to the corresponding iDEAL QR Transaction

API call. The iDEAL QR Transaction API response consists of a JSON message sent to the QR back-end by means of an HTTP 200 OK response.

The table below shows the fields that the Merchant must include in the iDEAL QR Transaction API response and the required format:

Name	Description	Format
`issuer_authentication_url`	The IssuerAuthenticationURL of the Issuing Bank, received in the iDEAL transaction response message from the Acquirer.	AN..max 512
`transaction_id`	The iDEAL TransactionID provided by the Acquiring Bank in the transaction response message.	PN..16

Fields in the iDEAL QR Transaction API response

After issuing the iDEAL QR transaction response to the iDEAL QR back-end, the Consumer will be redirected from the iDEAL App to the Issuer previously selected using the IssuerAuthenticationURL. The Consumer confirms and completes the transaction associated with the iDEAL QR code that has been scanned in the mobile banking environment of the Consumer's Issuing Bank. Upon completion, the Consumer will be sent back to the MerchantReturnURL provided by the Merchant in the iDEAL TransactionRequest. For this, the Merchant must follow the normal iDEAL protocol. This means that, after receiving a response from the Consumer, the Merchant must request the status and show this to the Consumer in the specified MerchantReturnURL.

6. Status after completion: iDEAL QR Status Protocol

Once the Consumer has been directed to the MerchantReturnURL, the iDEAL transaction has been completed. However, the final status of the iDEAL transaction relating to the iDEAL QR code must still be forwarded to and registered by the iDEAL QR back-end. To do this, the iDEAL QR back-end will request the status from the Merchant via an API call, which can be triggered in two ways:

1. Time: Every 30 minutes from the moment that the iDEAL QR back-end receives the iDEAL QR transaction response from the Merchant and only if this transaction has not yet received an iDEAL transaction end status.
2. One-off QR code scanned again: If a one-off QR code for which an iDEAL QR transaction request has been sent is scanned again, but only if this transaction has not yet received an iDEAL transaction end status.

6.1 iDEAL QR Status API call

Triggered by one of the two events mentioned above, the iDEAL QR back-end sends a Status API call to the Merchant. The Status API call consists of a JSON message that the Merchant sends to the earlier specified API URL by means of an HTTP POST.

The table belows shows the fields in the Status API call and the required format:

Name	Description	Format
`merchant_id`	The MerchantID associated with the iDEAL QR TransactionID at the iDEAL QR back-end.	PN..9
`merchant_sub_id`	The Merchant SubID associated with the iDEAL QR TransactionID at the iDEAL QR back-end.	N..Max 6
`transaction_id`	The iDEAL TransactionID known to the iDEAL QR back-end.	PN..16

Fields in the iDEAL QR Status API call

6.2 iDEAL QR Status response

In response to the Status API call, the Merchant forwards the iDEAL status to the iDEAL QR back-end, when the final status is known by the Merchant. If the final status is not yet known, the Merchant must respond with the status 'Open'. The Status API response consists of a JSON message that is sent to the QR back-end by means of an HTTP 200 OK response.

The table below shows the fields in the Status API call and the required format:

Name	Description	Format
`ideal_status`	The iDEAL status associated with the TransactionID, as provided in the Status API call. If the final status is not yet known, the Merchant must respond with status 'Open'.	AN..max 15

Fields in the iDEAL QR Status API response

One-off iDEAL QR codes for which the QR back-end has received the iDEAL status 'Success' will be locked and cannot be scanned again.

7. Error responses from back-end

In case of an error (e.g. validation, connection or authentication error) an HTTP 4xx or 5xx response with the error code and error message in the body is used (when possible) to communicate the error.

7.1 Error Response Message

The body of the message contains the element "status" which provdes the HTTP status code used, "code" which states the specific iDEAL QR error code and the "message" which states the applicable error message that corresponds to the iDEAL QR error code.

Fields in the iDEAL QR Error response:

Name	Description	Format
`status`	HTTP status code used for this error.	N. 3
`code`	The code that corresponds with a certain type of error.	N. 4
`message`	A message is provided that explains the error.	AN.

7.2 Error Codes

The following Error codes are currently available and may be used:

Error Code	Error message	Examples of providing error	HTTP Status Code
1001	Error while saving to the database	General internal error within back-end while trying to process request	500
1002	Record was not found in the database	iDEAL QR ID not found in database(404); MerchantID not found in database (400)	404; 400
1003	HTTP verb is not allowed	In case other HTTP verb is used than POST	405
1004	HTTP request was invalid	General error for incorrect elements in HTTP	400
1005	HTTP request validation failed	Merchant_token not recognized;	400
1006	HTTP request not found	Response not matchable with HTTP request	404
9998	Technical Error	Internal Technical Error	500
9999	Unknown	Unknown error, shall be returned if any unexpected exception is thrown	500

The Error Codes 100x can be communicated to both Merchant and iDEAL app. See HTTP Status Codes column to see which HTTP status code is to be used for which error code.

8. Processing speed for iDEAL QR Transaction protocol

When the Consumer confirms the QR transaction in the app and is redirected to the Issuer, the waiting time should be minimised in order to provide an optimal user experience. A target time is therefore specified and a time-out is used for the intermediate protocol between the Merchant and the IDEAL QR back-end. This time-out is 10 seconds for the Consumer. That means that if the Consumer is not redirected to the Issuer by the Merchant within 10 seconds after confirming the transaction, the iDEAL QR app will display an error message after which the QR code can be scanned by the Consumer again at any time (including one-off codes). Any iDEAL transaction that has been initiated will then expire.

The QR back-end will therefore assume a time-out period of 9.5 seconds between sending of the QR Transaction API call and the response to this call from the Merchant. However, the target time is 3.0 seconds.

Communication of iDEAL QR Transaction API	Target time (seconds)	Time-out (seconds)
API call from QR back-end API response from Merchant	3.0	9.5

Guidelines for processing speed of iDEAL QR Transaction protocol

9. iDEAL QR Security

To ensure the security of the data connection and message traffic between the Merchant and the iDEAL QR back-end, the following elements and measures are in place:

Merchant Token

Upon registering, the Merchant receives a unique Merchant Token. The Merchant must include this token for identification and authentication purposes in a separate field in the Generate QR API call messages that it sends to the iDEAL QR back-end to generate iDEAL QR codes. The Merchant must keep the token secret at all times. In the event that the key is compromised, the Merchant must contact the Acquirer immediately.

Certificate for TLS connection

A secure TLS connection is required for communication between the Merchant and the iDEAL QR back-end. The Merchant must be in possession of and make use of a valid TLS/SSL certificate, issued by any trusted Certificate Authority, in order to establish this TLS data connection with the iDEAL QR back-end. The iDEAL QR back-end will also use such a TLS certificate to authenticate itself and establish the TLS connection. The public keys must not be exchanged or pinned in advance. TLS version 1.1 or above must be supported for the exchange of messages between the Merchant and the iDEAL QR back-end.

Signing secret key

Upon registration, a unique secret signing key will be exchanged with the Merchant via the iDEAL QR back-end. The iDEAL QR back-end will sign all its API call and response messages through HMAC (SHA256). The body of the JSON message and the provided secret key are used as input variables for the HMAC function. The resulting HMAC hash is placed as signature in the message header under header name x-ideal-qr-hash. The merchant validates this signature through the same HMAC function and input variables.

The secret key must remain secret at all times. In the event that the key is compromised, the Merchant must contact the Acquirer immediately.

HMAC Example (to validate implementation):

Message body: `{"qr_id":"1234test","qr_url": "https://qrcode.ideal.nl/1234test"}`
Secret key: *key123*
Resulting HMAC signature hash: ae36cd6aeea48c050c3cf80f8bc25170f37fc2346d1ee294a8b815a2cca9c736

APPENDIX: Example iDEAL QR messages (Body)

iDEAL QR Generate API call

```
POST /ideal-qr/v1.0/generate
{
"merchant_token": "784aea4c-e36c-4a4b-b164-f9818aaeaf5c",
"merchant_sub_id": 5,
"amount": 24.95,
"amount_changeable": true,
"amount_max": 30.00, "amount_min": 20.00,
"description": "Product Y",
"one_off": true,
"expiration": "2016-05-14 00:00",
"beneficiary": "Organisatie X",
"purchase_id": "PO1234567",
"size": 1000
}
```

iDEAL QR Generate response

```
{
"qr_id": "5d6b159b-41ab-48eb-b379-da18ddea06dc",
"qr_url": https://qrcode.ideal.nl/codes/5d6b159b-41ab-48eb-b379-da18ddea06d?size=1000
}
```

iDEAL QR Transaction API call

```
POST [https://api.organizationx.nl/ideal-qr/transaction]
{
"merchant_id": 100298765,
"qr_id": "5d6b159b-41ab-48eb-b379-da18ddea06dc",
"issuer_id": "NLABC123",
"merchant_sub_id": 5,
"amount": 10.00,
"purchase_id": "PO1234567",
"description": "Product Y",
}
```

iDEAL QR Transaction response

```
{
"issuer_authentication_url": "https://www.bankx.nl/nl/ideal-
betalen/index.html?randomizedstring=123456789&trxid=002000123456789",
"transaction_id": "002000123456789",
}
```

iDEAL QR Status API call

```
POST [https://api.organizationx.nl/ideal-qr/status]
{
"merchant_id": 100298765,
"merchant_sub_id": 5,
"transaction_id": "002000123456789"
}
```

iDEAL QR Status response

```
{
```

```
"ideal_status": "success"
}
```

iDEAL QR Error response

```
{
"status": 405,
"code": 1003,
"message": "HTTP verb is not allowed"
}
```