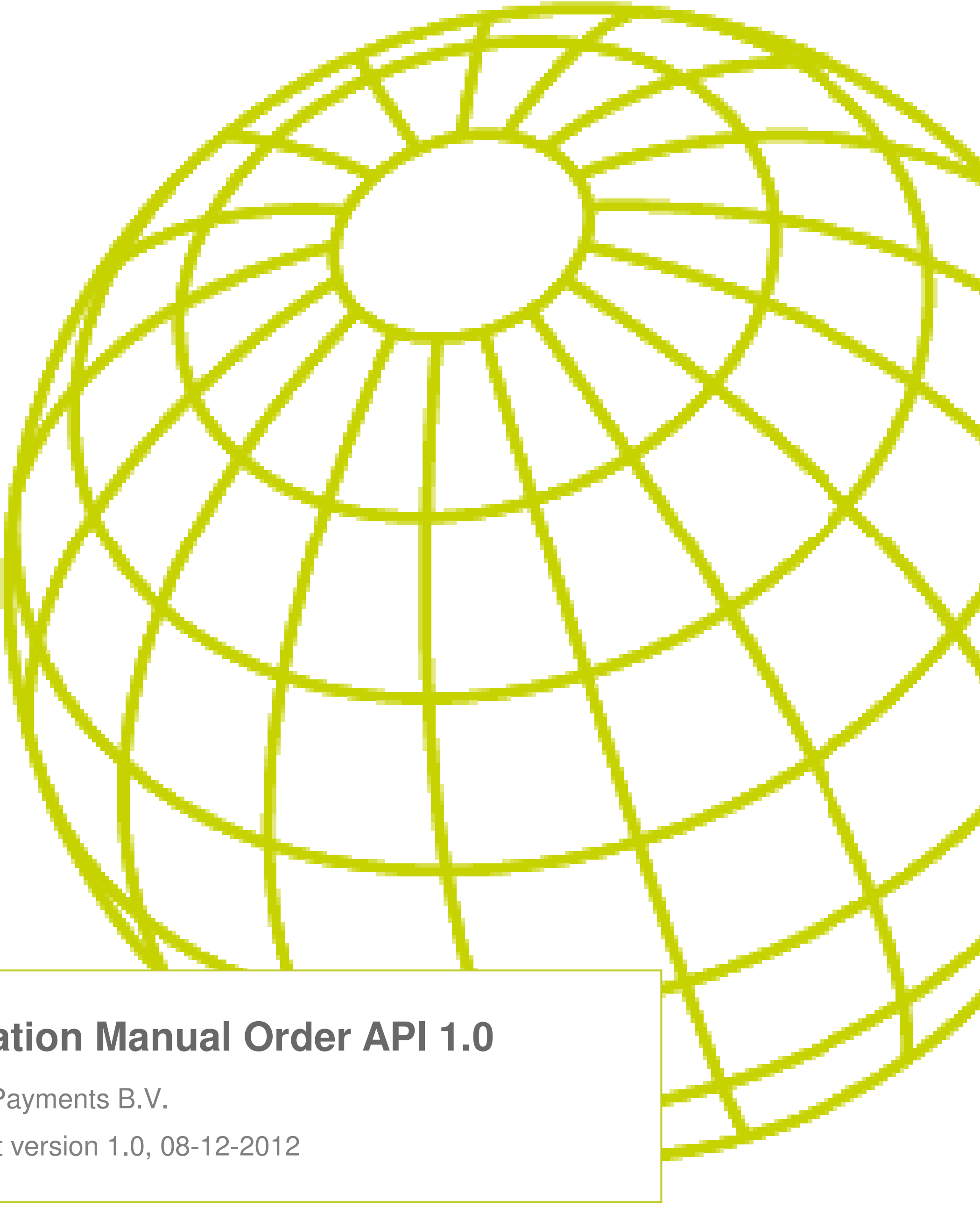


Integration Manual Order API 1.0

Docdata Payments B.V.

Document version 1.0, 08-12-2012

Introduction

This document describes the integration of the Docdata Payments Order API with merchants. The purpose of the integration is to communicate order and payment information to the Docdata Payments payment service.

The Docdata Payments Order API is a webservice created by Docdata Payments which can receive and send SOAP messages from and to the merchant concerning orders and their corresponding payments. It can be used to communicate the creation of a new order (to the payment service), to start a payment (on an order), request captures or refunds on a payment, to cancel an order and its payments or to request the current status of an order and its payments.

The Order API has been build to replace and improve upon the Command API. For a list of supported payment methods, please contact Docdata Payments Customer Services at support@docdatapayments.com, or by submitting a ticket on https://support.docdatapayments.com.

This manual describes version 1.0.

Table of contents

1. Integration steps

The following steps give a global idea of what will be required to integrate the Docdata Payments Order API into a service provider's current system:

- Use the supplied WSDL to create SOAP functionality for your current system
- Create functionality for collecting order (including shopping baskets) information and use this to make a call to the Order API's create command.
- (Optional and only after explicit approval of our support team) If not using Docdata's Payment menu, implement functionality for making a Web Direct payment using the Order API's start command.
- (Optional) If not making use of Docdata Payments automatic capture setting, implement functionality for submitting capture requests using the Order API's capture command. Alternatively, this can be done through Docdata Payments back office by using the capture delay option.
- (Optional) Implement functionality for submitting refund request using the Order API's refund command. Alternatively, this can also be done through Docdata Payments back office.
- (Optional) Implement functionality for canceling a previously created order and its payments (if any were made) using the Order API's cancel command. Alternatively, this can also be done through Docdata Payments back office.
- Implement functionality for requesting the status of an order and its payments using the Order API's status command.
- Implement functionality for receiving status change notifications.

2. Points of interest for implementation

2.1 Changes from the Command API

For those who are transitioning from the Command API, please take notice of the following:

- SOAP – In the Command API requests were made with HTTP GET commands, now all communication is done through SOAP HTTP.
- Payment methods – The payment methods available in the Order API may be different from those in the Command API.
- Merchant password always required – In the Command API the merchant password was not required for every call. For security reasons this must now be supplied with every call.
- Separate capture – In the Command API there was not a dedicated call for making captures.
- Payment always separate – In the Command API (recurring) payments could be made while creating a new payment order cluster. Payments now should always be performed through the start command or the web menu.
- Multiple capture/refund requests – In the Command API it was not possible to perform multiple captures or refunds. It is now possible through the new Order API. For refunds multiple partial refunds are enabled. Multiple partial capture is expected in the course of 2013.
- Merchant references – In the Order API a merchant may supply merchant reference IDs when creating orders, captures or refunds, which the merchant can use for identification purposes
- Capture/Refund descriptions – In the Order API a merchant may supply descriptions for captures or refunds
- Added order items – In the Order API it's possible to specify order item details, including an optional link to the item image
- Single status report – The Order API now always returns the same kind of status report
- Optional new web menu – A new payment menu has been developed which uses the new functionality made available through the Order API (see chapter 3). Merchants can remain on the old web menu while still making use of the new Order API
- Shoppers who leave the menu no longer automatically cancel the order. The

merchant now can control if and when the order is cancelled

2.2 Changes from Order API version 0.9

For those who are transitioning from Order API version 0.9, please take notice of the following:

- Recurring payments – It is now possible to create recurring payments through the Order API. The create and start operations are extended with a new parameter that can link to other Payments in a recurring theme. For more information see chapter 1.9.
- Cancellation of captures and refunds – Captures and refunds with the NEW state can now be replaced by a new capture or refund. Once the status has changed to STARTED this is no longer possible.
- Refunds for SEPA bank accounts – It is now possible to refund payment methods which require a shopper bank account like MisterCash and Sofort. For more information see chapter 6.5.
- Starting Web Direct payments – Web Direct payments can now be started for Direct Debit, Bank Transfer and ELV. For more information see chapter 6.2.
- Extended status operation – A new web service operation has been added which includes payment method specific details. See chapter 7 for more information.
- Approximate totals – It is required by law to communicate exchange rates to shoppers. Therefore approximate totals are added to the status operation response. See chapter 7 for more information.
- Chargebacks in status operation – Besides captures and refunds, chargebacks are now also included in the standard status operation response. See chapter 7 for more information.

2.3 Status changed notification service

It is possible to use the status changed notification functionality of the Docdata Payments payment service to be notified when the status of a specific payment changes for the merchant by calling a specified URL located at the website of the merchant. The Docdata Payments payment service will execute the Status Changed Notification for every status change of every payment in an order. The URL will be called every time a state within an order changes.

The parameter will be the merchant order reference as supplied when creating an order. An example of the call from the Docdata Payments payment service will look as follows with a merchant order reference of 33:
http://www.merchantwebsite.com/update_order?order_id=33

The Status Changed Notification functionality is included in the Docdata Payments payment service back office. An example of how the call should look like is shown below:
http://www.merchantwebsite.com/update_order?order_id=

The Status Changed Notification will only be supported when the website of the merchant supports the GET operation and when merchant order reference can be added to the last parameter of the call. At the end of call of the return URL should be an ”=” sign. Furthermore, the call to the merchant’s website will be made on port 80 (HTTP) or port 443 (HTTPS), other ports are blocked due to safety reasons.

Important: It is possible that the notification of the Docdata Payments payment service Status Change Notification does not reach the merchant due to a network failure. In that case, the notification will be sent seven times, three times a day, when no HTTP status code 200 is read.

3. Interfacing through SOAP

The Order API communicates solely using a SOAP interface over HTTP. SOAP allows the Order API to make a stronger definition of the required input and output for its different operations. This definition is described in a WDSL document, which can be easily exchanged with the merchant, who can use this to easily implement the required SOAP functionality. Being HTTP calls, the commands are synchronous and thus will always return an answer. Some operations however such as capture or refund request will only confirm that the request has been successfully created, and the merchant will receive a notification on a later time asynchronously when the request has been processed.

The SOAP response message for each call will always contain either an “error” element containing an error message indicating what went wrong, or a “success” element which can contain the call specific information or be empty.

The Order API is accessible as a public service, though each Order API command requires a merchant name and password for authentication.

For more information on SOAP, please refer to W3C's official documentation on their website: http://www.w3.org/

The WSDL of this version of the Order API and information concerning the web service address can be found on the following locations:
Testing: https://test.docdatapayments.com/ps/services/paymentservice/1_0
Production: https://secure.docdatapayments.com/ps/services/paymentservice/1_0

4. Web menu

Independent of the new Order API 1.0 a new web menu has been developed for the payment of your orders. This new web menu is responsible for handling the payment process for web shops and provides a lot of added functionality compared to the existing web menu which merchants are currently using.

When the shopper has finished making his/her payment in Docdata Payments web menu, Docdata Payments can use the status changed notification service as described in chapter 2.3 to notify the merchant of the change in the order data. The use of this service is optional, but recommended.

For information on implementing Docdata Payments web menu, please refer to the web menu's official documentation.

4.1 Impact on the new web menu when using Order API

With the new Order API merchants have the possibility to indicate the items linked to the payment order. The new web menu can make use of this data when it's available to display the order items and their information, instead of just presenting the shopper with a total price. The following order item properties can be displayed:

- Name
- Description
- Price calculations
- Image

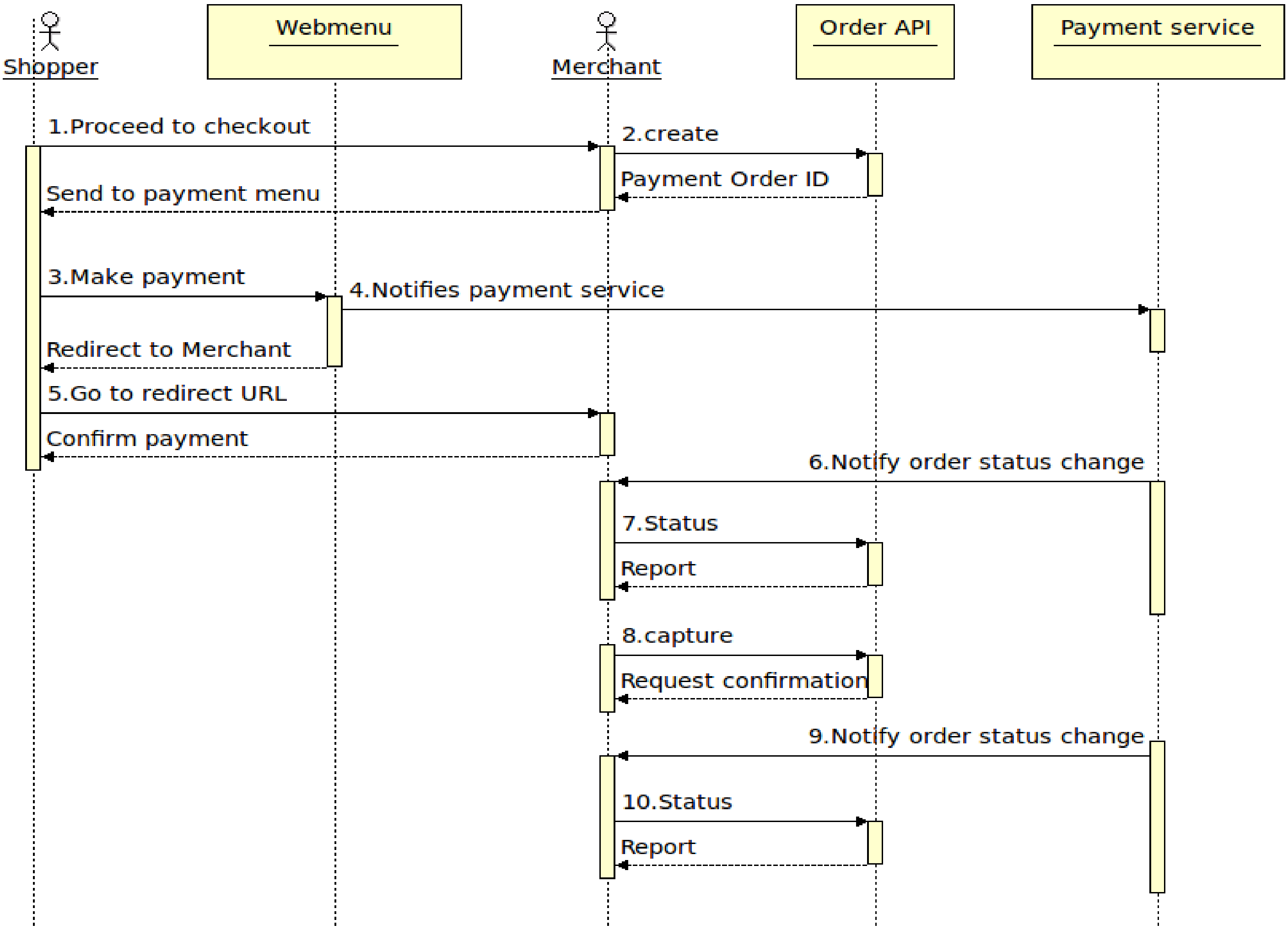
Another functionality the new web menu allows only in combination with the Order API is the option to use the Afterpay payment method, where the shopper pays on receiving the ordered goods / services.

5. Process overview

This chapter will explain the use of the Order API in different common scenarios from the perspective of the merchant.

5.1 Normal order using web menu

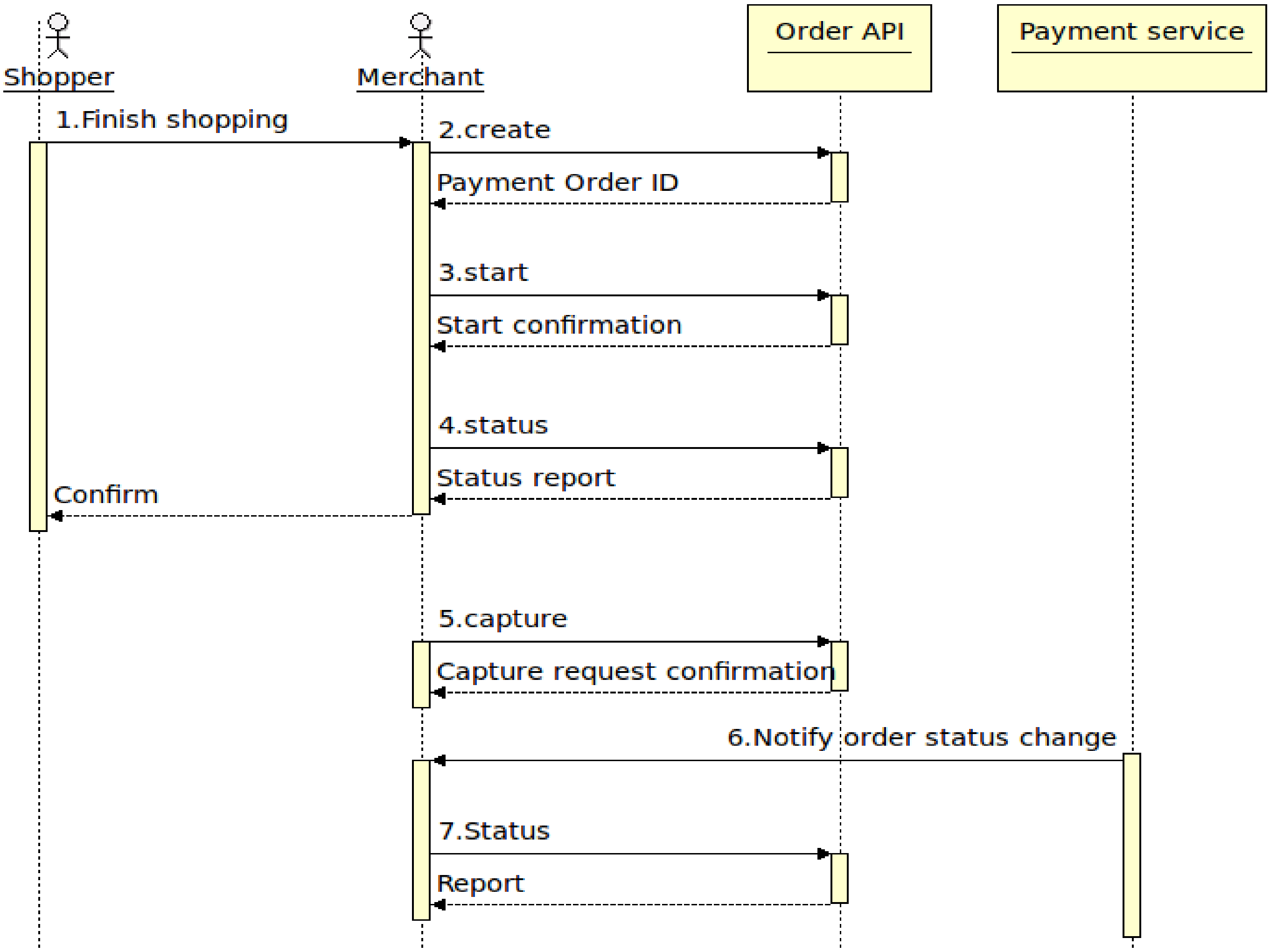
This workflow shows the different steps in which an order and its payments is processed through use of the Order API and the web menu.



1. Shopper has finished shopping and proceeds to checkout, after which he is sent by the Merchant to the payment menu.
2. The Merchant creates a Payment Order for this user using the Order API
3. Shopper processes his/her payment in the web menu after which he/she is redirected back to the merchant's web shop.
4. web menu informs Docdata Payments payment service of the payment made.
5. Shopper goes back to the merchant's web shop via the web menu's redirect URL
6. Docdata Payments payment service notifies the merchant of the change in the order status.
7. Merchant checks the status of the order to see whether the payment has been authorized or declined.
8. (Optional) When the Merchant ships the sold goods or services, the required captures are performed. This will not be necessary if the merchant has chosen to set the option to automatically perform a capture for the full order amount.
9. After the capture has been processed, Docdata Payments payment service notifies the merchant of a change in the payment.
10. Merchant checks the status to see whether all payments have been completed successfully.

5.2 Normal order using Web Direct

This workflow shows the typical steps for processing an order using the Web Direct functionality for handling the payment.



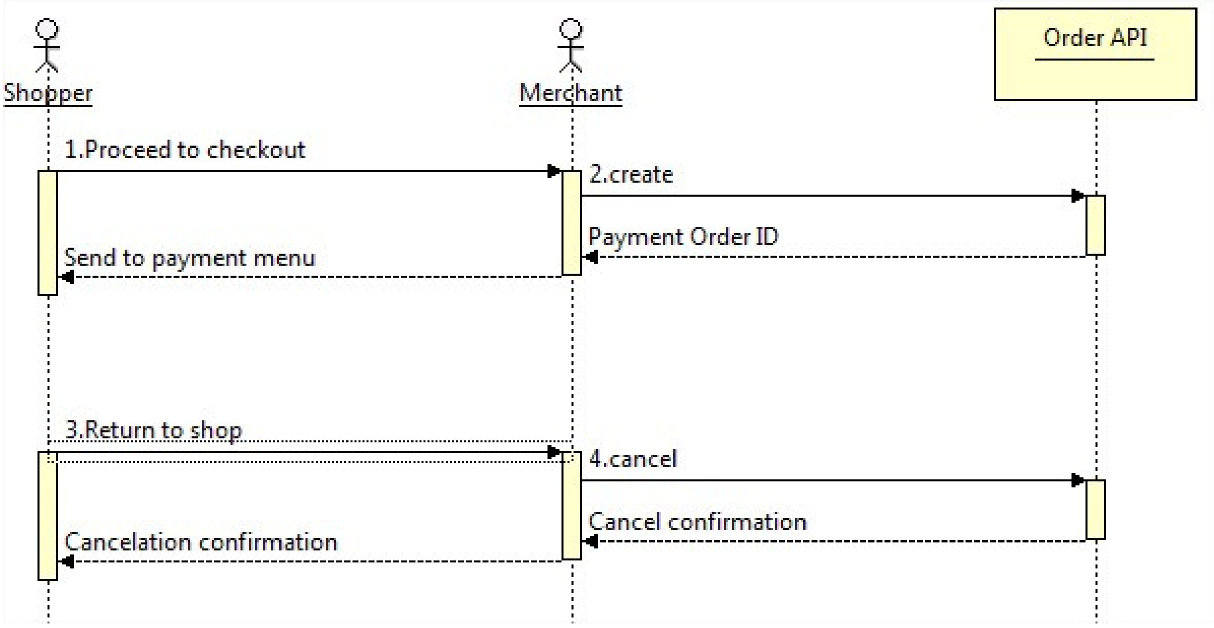
1. The Shopper has finished shopping and processes payment in the Merchant's payment menu.
2. The Merchant creates a Payment Order for this user using the Order API.
3. The Merchant starts the payment of the created order.
4. The Merchant checks whether the payment has been authorized successfully.
5. (Optional) When the Merchant ships the sold goods or services, the requests for capture can be performed. This will not be necessary if the merchant has chosen to

set the option to automatically perform a capture for the full order amount.

6. Docdata Payments payment service informs merchant of order status change.
7. Merchant checks the status to see whether all captures have succeeded.

5.3 Payment cancellation process

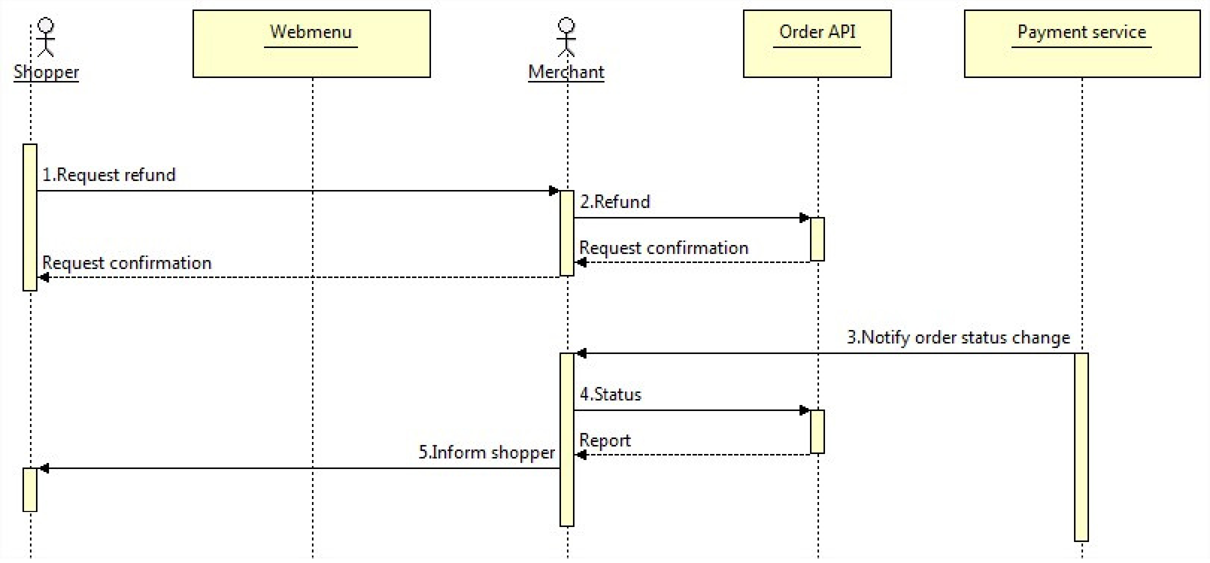
This workflow shows the typical steps for creating a Payment Order and immediately canceling it.



1. Shopper is done shopping and proceeds to checkout, after which he or she is sent to the merchant's payment menu
2. The Merchant creates a Payment Order for this user using the Order API
3. The Shopper leaves the menu and returns to the shop
4. Merchant cancels the Payment Order at the Order API

5.4 Refund process

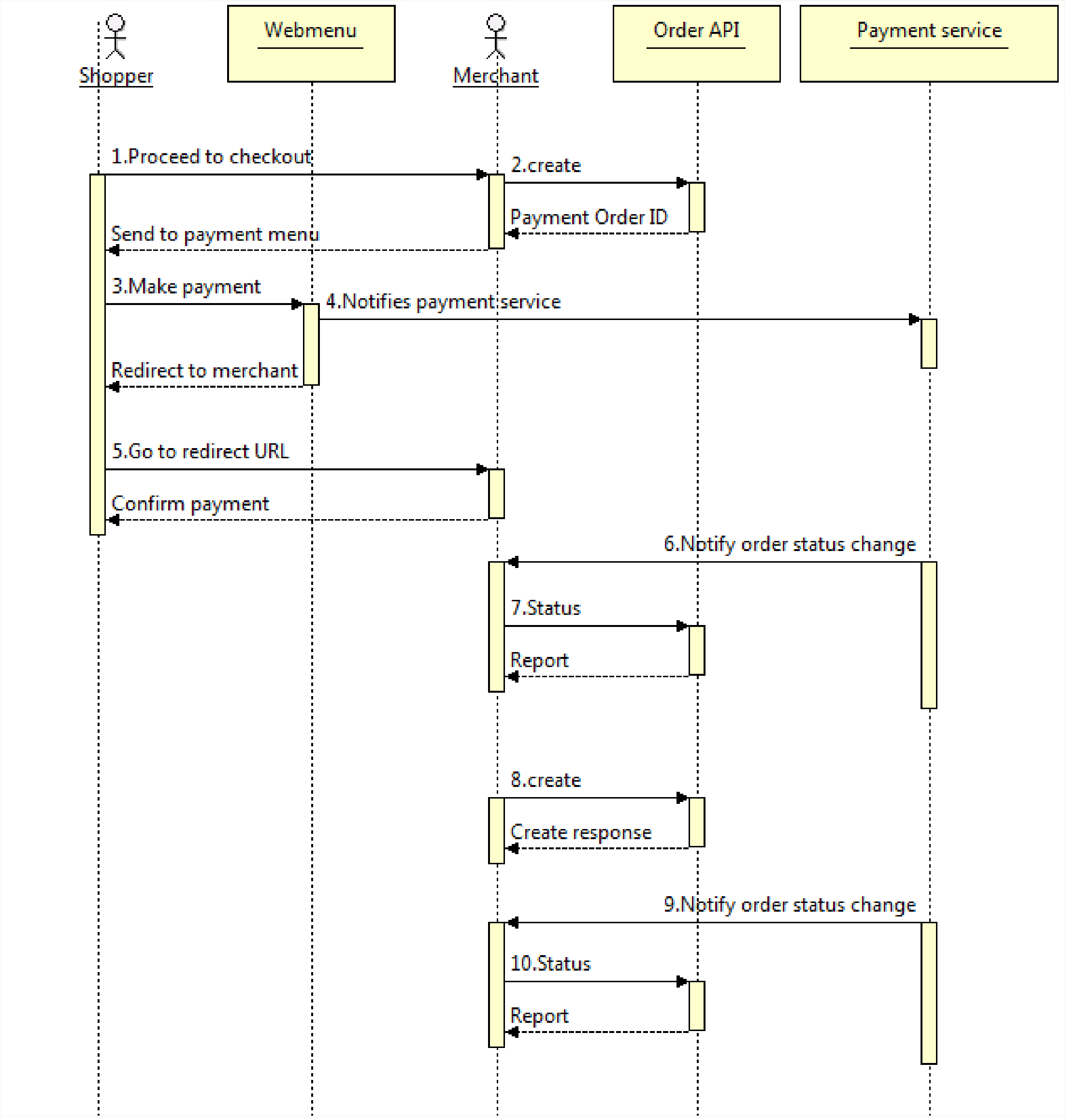
This workflow shows the typical steps for a refund process.



1. Shopper is done shopping and proceeds to checkout, after which he/she is sent to the payment menu.
2. The Merchant creates a Payment Order for this user using the Order API.
3. Shopper finishes up his/her payment in the web menu after which the web menu sends him/her to a merchant result URL.
4. Web menu informs Docdata Payments payment service of the payment made.
5. Shopper goes back to the merchant's web shop via the web menu's redirect URL. The merchant can use this as notification that the payment has been successfully started.
6. Docdata Payments payment service notifies the merchant of the change in the order status.
7. Merchant checks the status of the order to see whether the payment has been authorized.
8. (Optional) When the Merchant ships the sold goods or services, the requests for capture can be performed. This will not be necessary if the merchant has chosen to set the option to automatically perform a capture for the full order amount.
9. After the capture has been processed, Docdata Payments payment service notifies the merchant of a change in the payment.
10. Merchant checks the status to see whether all payments have been completed successfully.
11. Shopper requests a refund.
12. Merchant sends the refund request to the Order API.
13. Docdata Payments payment service informs merchant of a change in the payment status.
14. Merchant requests the payment status to check whether the refund has been processed.
15. Merchant informs shopper.

5.5 Recurring Order

This workflow shows the typical steps for a recurring order.



1. Shopper is done shopping and proceeds to checkout, after which he/she is sent to the payment menu
2. The Merchant creates the *initial recurring payment order* for this user using the Order API by supplying a new unique payment reference in the create call.
3. Shopper finishes up his/her payment in the web menu after which the web menu sends him/her to a merchant result URL
4. Web menu informs Docdata Payments payment service of the payment made
5. Shopper goes back to the merchant's web shop via the web menu's redirect URL. The merchant can use this as notification that the payment has been successfully started
6. Docdata Payments payment service notifies the merchant of the change in the order status
7. Merchant checks the status of the order to see whether the payment has been authorized
8. Merchant creates a recurring order using the Order API by supplying the unique payment reference used in step 2 in the create call. This will automatically creates the recurring payment based on the initial recurring order's payment.
9. Docdata Payments payment service notifies the merchant of the change in the order status
10. Merchant checks the status of the order to see whether the recurring payment has been authorized

Note that it's also possible to create the initial recurring order using web direct. In this case the creation of the initial order creation remains the same, but for the subsequent orders the payment reference should be supplied with the *start* call instead of the *create* call. So in other words, the subsequent order is created just like any order, but when starting the order's payment it uses the payment reference to make use of the initial order's payment details.
It's also possible to combine manual capturing with the recurring order workflow, in which case the captures work exactly as they would in a normal payment order workflow.

6. Docdata Order API Operations

This chapter describes the different operations available in the Order API and their use. As previously mentioned, the Order API can only be called through the use of SOAP messages. For a full description of every parameter and their constraints, please refer to the WSDL. For a readable version of the WSDL, please refer to the PDF for the Order API WSDL document that is available from Docdata Payments.

6.1 Create

The goal of the create operation is solely to create a payment order on Docdata Payments system. Creating a payment order is always the first step of any workflow in Docdata Payments payment service.

After an order is created, payments can be made on this order; either through (the shopper via) the web menu or through the API by the merchant. If the order has been created using information on specific order items, the web menu can make use of this information by displaying a shopping cart.

6.1.1 Request

A quick overview of different groups of parameters which can be used in the create call:

- Merchant credentials – Required for each call
- Merchant order reference – The merchant's internal reference ID for this order. This will be used by Docdata Payments when notifying merchants of an order status update (see chapter 2.3)
- Payment preferences – Specific settings to use for all payments which are going to be made on this order, for example the payment profile ID
- Menu preferences – Preferences to be used by the Docdata Payments web menu. Please check which stylesheetID you pass along in your XML, as the new menu requires a separate stylesheet.
- Shopper – Information about the shopper who made the order
- Item – Information about one or more of the order items. Can be used by the Docdata Payments web menu to display order item information or by capture/refund to

determine the capture/refund amount.

- TotalGrossAmount/TotalNetAmount/TotalVatAmount – Total amounts and their respective currencies
- Bill to / Ship to – Billing/Shopping destinations
- Description / Additional description – A description for this order, and an additional description if the normal description is insufficient
- Payment Request – A reference to a previously made payment, to be used when making a recurring order. To make the initial recurring order, the merchant needs to supply a new unique reference identifier here.

6.1.2 Response (normal)

The response of the create call will contain either an error or a success:

- Error: If no order could be created with the given request, this will contain an error message specifying what went wrong
- Success: If an order was successfully created, this will contain the payment order key which will need to be used for starting payments, requesting the order status or for canceling the order.

  If a payment request has been used in the request, the success in the response will indicate one of three possible responses:

  - paymentSuccess - A payment has been successfully made for this order. This is the usual response for a subsequent recurring order.
  - paymentError - A payment has failed for this order. This is a possible response for a subsequent recurring order, for instance when the initial order's payment details are not valid anymore (expired credit card).

6.1.3 Response (recurring)

When the order is created with a payment request, e.g. in the case of recurring orders the success response can also contain a third option.

- Error: If no order could be created with the given request, this will contain an error message specifying what went wrong
- Success: If an order was successfully created, this will contain the payment order key which will need to be used for starting payments, requesting the order status or for canceling the order.

If a payment request has been used in the request, the success in the response will indicate one of three possible responses:

- paymentInsufficientData - No data has been found to make the payment, meaning one will have to be made through the web menu or web direct. This is the usual response when creating the initial recurring order.
- paymentSuccess - A payment has been successfully made for this order. This is the usual response for a subsequent recurring order.
- paymentError - A payment has failed for this order. This is a possible response for a subsequent recurring order, for instance when the initial order's payment details are not valid anymore (expired credit card).

6.2 Start

The start operation is used for starting a (web direct) payment on an order. It does not need to be used if the merchant makes use of Docdata Payments web menu.

6.2.1 Request

The following can be used when making a request on the start call:

- Merchant credentials – Required for each call
- Payment Order Key – The key that was returned from a successful create call, used to identify for which order the payment has been made.
- One of
  - Payment – The information required for making the actual payment, see below.
  - Recurring payment request – A reference to a previous payment to indicate those payment details will need to be used for making this recurring payment.

Payment information

Some payment information is needed to start a web direct payment. The chosen payment method needs of course to be configured in the merchant’s profile.

- Payment method – Name of the payment method to be used. See list of supported methods below.
- Payment amount – Optional payment amount. If left empty, the full amount of the payment order is used.
- One of the payment-specific input groups matching the aforementioned provided

payment method name. If e.g. BANK_TRANSFER as payment method name (see list below) is used, the `bankTransferPaymentInput` element is expected to be provided.

The following names of payment methods (and associated input elements) are supported for the start operation:

Payment method name	**Required input element**
AMEX	`amexPaymentInput`
MASTERCARD	`masterCardPaymentInput`
VISA	`visaPaymentInput`
DIRECT_DEBIT	`directDebitPaymentInput`
BANK_TRANSFER	`bankTransferPaymentInput`
ELV	`elvPaymentInput`

Supported payment methods web direct starting

While the parameters per payment-specific input element closely resemble the information requested of the shopper in the web menu, the API sometimes asks for more details. E.g. for direct debit, while shopper has to enter just an account number if paying through the web menu, merchant needs to provide complete IBAN and BIC instead.

Recurring payment request

See for more information about recurring orders chapter 5.5

6.2.2 Response

The response of the start call will contain either an error or a success:

- Error: If no payment could be started with the given request, this will contain an error message specifying what went wrong
- Success: If a payment was successfully started, this will contain the payment id which will need to be used for making captures or refunds. The payment ID can also be obtained by requesting the status for a certain order

6.3 Cancel

The cancel command is used for canceling a previously created payment, and can only be used for payments with status NEW, STARTED and AUTHORIZED.

6.3.1 Request

The input parameters of the cancel request consist solely of the following parameters:

- Merchant credentials – Required for each call
- Payment Order Key – The key which was returned in a successful create call, used to identify the order which needs to be canceled

6.3.2 Response

The cancel response will simply be an error containing an error message, or a success indicating that the order was canceled and all attached payments were reverted.

6.4 Capture

The capture command is used to create requests for performing captures on authorized payments. A merchant can choose to have it set up through Docdata Payments back office to automatically have the full authorization amount captured for each payment after a configured delay. The capture command can then be used to overwrite this default capture. If no default capture is configured, a merchant should use the capture command to create one.

6.4.1 Request

The capture command request contains the following parameters:

- Merchant credentials – Required for each call
- Payment ID – The ID of the payment on which a capture is requested. This ID is returned by the start command's response, and is also listed in the status response
- Merchant capture reference – A merchant's own reference for this capture. Can be used to identify a capture when it's being returned with a status call
- Amount – The amount to capture. If the amount and the item code is left empty, the total capturable amount left on the authorization will be used
- Item code – An item code which was given during the create request call. When an item code is supplied instead of an amount, the item's total gross amount is used instead. If both an amount and an item code are supplied, the amount will be used
- Description – Description of the capture
- Final capture – A boolean indicator for stating that this will be the last capture made on this payment
- Cancel reserved - A boolean indicator for stating that this capture should replace a default capture, if one exists
- Required capture date - A date indicating when the capture should be performed.

6.4.2 Response

The capture request command does not directly make captures itself, it merely creates a capture request to be processed. Therefore, the capture response will only indicate whether the request has been successfully created or not. For more information on whether the capture itself has been processed successfully or not the status command will have to be used.

This is why the capture response is either an error containing an error message, or a success indicating the capture request was filed successfully.

6.5 Refund

The refund command is used to create requests for performing one or more refunds on payments that have been captured successfully. Its functionality is very similar to submitting captures.

6.5.1 Request

Since the refund command is similar to the capture command, its request consists of similar parameters:

- Merchant credentials – Required for each call
- Payment ID – The ID of the payment on which a refund is requested. This ID is returned by the start command's response, and is also listed in the status response
- Merchant refund reference – A merchant's own reference for this refund. Can be used to identify a refund when it's being returned with a status call
- Amount – The amount to refund. If the amount and the item code is left empty, the total refundable amount left on the authorization will be used
- Item code – An item code which was given during the create request call
- When an item code is supplied instead of an amount, the item's total gross amount is used instead. If both an amount and an item code are supplied, the amount will be used
- Description – Description of the refund
- Refund bank account – Optional parameter used for refunding payments where additional information is needed about the bank account.
- Cancel reserved - A boolean indicator for stating that this refund should replace previously created, not yet executed, refunds
- Required refund date - A date indicating after which the refund should be performed.

6.5.2 Response

Like the capture response, the refund response will only indicate whether the request has been filed successfully or not. To get more information on the refund status, the status command has to be used.
The refund response will either be an error containing an error message, or a success indicating that the refund request has been filed successfully.

6.6 Status

The status call can be used to get a report on the current status of a Payment Order, its payments and its captures or refunds. It can be used to determine whether an order is considered "paid" (see chapter 6 for more information on this), to retrieve a payment ID, to get information on the statuses of captures/refunds. Since Order API version 1.0 there is also an extended status operation that requires the same input parameters as the regular status operation but returns payment method specific data in the response.

6.6.1 Request

The status request is rather simple; it only requires the following parameters:

- Merchant credentials – Required for each call
- Payment Order Key – The key which was returned in a successful create call, used to identify the order for which the status report is requested

6.6.2 Response

The response of the status call will either be an error, or a success containing the status report the order. For more information about interpreting the (extended) status report, please refer to chapter 7.

7. Interpreting status report

The status report can return a lot of information concerning an order and its payments, not all of which may be relevant for every merchant. This chapter aims to explain the different information that the status call returns to help determine which information to use. For more detailed information, please see the WSDL and XSD of the SOAP service.

7.1 Status report example XML

Authorized payment

The status message consists of a status element *(<success>)* and the actual report *(<report>*). The status element only indicates that the status operation was successful, it does not indicate anything about the order itself.

The report element contains an *<approximateTotals>* element and *<payment>* elements for each payment. The totals are a summary of the status of this order and can be used to easily determine whether or not to deliver the goods. For a discussion about how to determine that, see "Determining whether an order is paid". Note that these totals are approximations: amounts are converted to your order's currency and since exchange rates may vary, they can only be used as an indication.

The payment elements contain a detailed description of each payment (attempt) made by the shopper. If you need more control then the totals provide, you can inspect these elements to have a more fine grained insight.

```
<?xml version='1.0' encoding='UTF-8'?>
<statusResponse xmlns="http://www.docdatapayments.com/services/paymentservice/1_0/">
  <statusSuccess>
    <success code="SUCCESS">Operation successful.</success>
    <report>
      <approximateTotals exchangedTo="EUR" exchangeRateDate="2012-12-04 14:39:53">
        <totalRegistered>3310</totalRegistered>
        <totalShopperPending>0</totalShopperPending>
        <totalAcquirerPending>0</totalAcquirerPending>
        <totalAcquirerApproved>3310</totalAcquirerApproved>
        <totalCaptured>0</totalCaptured>
        <totalRefunded>0</totalRefunded>
        <totalChargedback>0</totalChargedback>
      </approximateTotals>
      <payment>
        <id>1606709142</id>
        <paymentMethod>MASTERCARD</paymentMethod>
        <authorization>
          <status>AUTHORIZED</status>
          <amount currency="EUR">3310</amount>
          <confidenceLevel>ACQUIRER_APPROVED</confidenceLevel>
        </authorization>
      </payment>
    </report>
  </statusSuccess>
</statusResponse>
```

Cancelled payment

If a payment attempt failed, e.g. by the shopper cancelling the attemp, providing wrong details or by not having enough credit, the payment is assigned status *CANCELED*. No confidence level is assigned.

```
<?xml version='1.0' encoding='UTF-8'?>
<statusResponse xmlns="http://www.docdatapayments.com/services/paymentservice/1_0/">
    <statusSuccess>
        <success code="SUCCESS">Operation successful.</success>
        <report>
            <approximateTotals exchangedTo="EUR" exchangeRateDate="01-08-2012 12:12:12">
                <totalRegistered>3310</totalRegistered>
                <totalShopperPending>0</totalShopperPending>
                <totalAcquirerPending>0</totalAcquirerPending>
                <totalAcquirerApproved>0</totalAcquirerApproved>
                <totalCaptured>0</totalCaptured>
                <totalRefunded>0</totalRefunded>
            </approximateTotals>
            <payment>
                <id>2162274147</id>
                <paymentMethod>DIRECT_DEBIT</paymentMethod>
                <authorization>
                    <status>CANCELED</status>
                    <amount currency="EUR">3310</amount>
                    <confidenceLevel> </confidenceLevel>
                </authorization>
            </payment>
        </report>
    </statusSuccess>
</statusResponse>
```

Payment with captures, refund and chargebacks

After a payment is authorized it will be captured. While the authorization is merely a reservation, the capture is the actual indication of money being transfered. Shoppers can also be refunded (by the initiative of the merchant) or they can claim a chargeback (initiated by the shopper). These further steps in the lifecycle of a payment are noted in separate *<capture>*, *<refund>*, and *<chargeback>* elements underneath the *<authorization>* element.

```
<statusResponse xmlns="http://www.docdatapayments.com/services/paymentservice/1_0/">
   <statusSuccess>
     <success code="SUCCESS">Operation successful.</success>
     <report>
       <approximateTotals exchangedTo="EUR" exchangeRateDate="2012-12-07 17:46:19">
         <totalRegistered>3310</totalRegistered>
         <totalShopperPending>0</totalShopperPending>
         <totalAcquirerPending>0</totalAcquirerPending>
         <totalAcquirerApproved>3310</totalAcquirerApproved>
         <totalCaptured>3310</totalCaptured>
         <totalRefunded>3310</totalRefunded>
         <totalChargedback>3310</totalChargedback>
       </approximateTotals>
       <payment>
         <id>1606707559</id>
         <paymentMethod>MASTERCARD</paymentMethod>
         <authorization>
           <status>AUTHORIZED</status>
           <amount currency="EUR">3310</amount>
           <confidenceLevel>ACQUIRER_APPROVED</confidenceLevel>
           <capture>
             <status>CAPTURED</status>
             <amount currency="EUR">3310</amount>
           </capture>
           <refund>
             <status>CAPTURED</status>
             <amount currency="EUR">3310</amount>
           </refund>
           <chargeback>
```

```
            <chargebackId>24</chargebackId>
            <status>CHARGED</status>
            <amount currency="EUR">3310</amount>
            <reason>automatic chargeback</reason>
          </chargeback>
        </authorization>
      </payment>
    </report>
  </statusSuccess>
</statusResponse>
```

7.2 Extended status report example XML

Extend status reports are only available for certain merchants. Ask your sales contact for more information. The extended status adds an `<extended>` information element which provides payment method specific details which may be of interest for more advanced integrations. The `extended` element is currently only available for Amex, Mastercard, VISA, Maestro, Mistercash, iDEAL, Bank Transfer and Gift Cards. In future versions, more details can be added. See the WSDL and XSD descriptions for specific details per payment method.

Authorized Payment

```
<?xml version='1.0' encoding='UTF-8'?>
<extendedStatusResponse
xmlns="http://www.docdatapayments.com/services/paymentservice/1_0/">
  <statusSuccess>
    <success code="SUCCESS">Operation successful.</success>
    <report>
      <approximateTotals exchangedTo="EUR" exchangeRateDate="2012-12-04 15:16:56">
        <totalRegistered>3310</totalRegistered>
        <totalShopperPending>0</totalShopperPending>
        <totalAcquirerPending>0</totalAcquirerPending>
        <totalAcquirerApproved>3310</totalAcquirerApproved>
        <totalCaptured>0</totalCaptured>
        <totalRefunded>0</totalRefunded>
        <totalChargedback>0</totalChargedback>
      </approximateTotals>
      <payment>
        <id>1606709142</id>
        <paymentMethod>MASTERCARD</paymentMethod>
        <authorization>
          <status>AUTHORIZED</status>
          <amount currency="EUR">3310</amount>
          <confidenceLevel>ACQUIRER_APPROVED</confidenceLevel>
          <capture>
            <status>NEW</status>
            <amount currency="EUR">3310</amount>
          </capture>
        </authorization>
        <extended>
          <riskChecks score="100">
             <check score="100">3ds.merchant.liability</check>
             <check score="0">3ds.issuer.liability</check>
          </riskChecks>
          <masterCardPaymentInfo>
            <cardHolder>John Doe</cardHolder>
            <cardCountry>US</cardCountry>
            <panLast4>9943</panLast4>
          </masterCardPaymentInfo>
        </extended>
      </payment>
    </report>
  </statusSuccess>
```

7.3 Explanation of terms

- Acquirer – The bank or financial institution that processes payments for a merchant
- Totals – An overview of the totals amounts which have been authorized, captured or are still pending
- Total Registered – The total amount in the currency of the order, which was listed by the merchant in the request of the create command under “TotalGrossAmount”. If order currency and payment currency differ, the totals will be converted using the daily exchange rates of the European Central Bank (ECB)
- Total Shopper Pending – The total amount which the shopper has promised to pay, for instance through a manual bank transfer. It's the total sum of all authorization amounts with the confidence level “SHOPPER_PENDING”
- Total Acquirer Pending – The total amount which needs to be paid through an acquirer, of which there is no guarantee that it will actually be capturable. It's the total sum of all authorization amounts with the confidence level “ACQUIRER_PENDING”
- Total Acquirer Approved – The total amount which the acquirer has promised to transfer to the merchant. It's the total sum of all authorization amounts with the confidence level “ACQUIRER_APPROVED”
- Total Captured – The sum of all completed captures, ie all captures with the state “CAPTURED”
- Total Refunded – The sum of all completed refunds, ie all refunds with the state “CAPTURED”
- Payment – An individual payment for this order, which consists of an ID, payment method and possibly an authorization
- Payment ID – The identifier of this payment. Used when making captures or refunds
- Payment method – The method which was used when making this payment, for example Mastercard, iDeal, Paypal, Giftcards, etc.
- Authorization – The approval of an acquirer or buyer to perform a payment for a certain amount. There can be at most one authorization per payment, and the authorization itself can contain a number of captures (actual deduction of funds from a credit card for instance) and refunds
- Authorization confidence level – Shows how much confidence there is that the capture of the authorized payment will actually succeed. Currently there are three possible levels of confidence:

- ACQUIRER_PENDING - Acquirer is asked to process the authorization, but there are some possibilities that the money won't be transferred. This can be shown for example when making a direct debit payment, where there is the possibility that there is not enough money to make the payment
    - SHOPPER_PENDING – Shopper has promised that the money can be captured, for example when the shopper will make a bank transfer. There is no guarantee if or when this could happen
    - ACQUIRER_APPROVED - Guaranteed by acquirer that the authorization can be captured. This can be shown for example with a credit card payment or iDEAL and Bancontact/Mister Cash
- Capture – A merchant request to collect a certain amount, most commonly used with credit card transactions. Methods like iDEAL and Bancontact/Mister Cash will see an immediate automatic capture after the authorization
- Refund – A merchant request to return a certain captured amount
- Merchant reference – The identifier which the merchant supplied when creating the capture or refund
- Status – The current status of the capture or refund request. Current possible statuses are:
    - NEW – Initial state; the request has only been filed
    - STARTED – Capture or refund is requested and grouped in a batch to be processed
    - CAPTURED – Capture or refund was processed successfully
    - CANCELED - Capture or refund has been revoked, either manually through a cancel order command, through another capture or through an external process
    - FAILED - Capture or refund was processed but declined by the acquirer
- ERROR - Capturing failed due to an error

7.4 Determining whether an order is paid

Different merchants can have different ways of determining when they consider an order “paid”, the totals in the status report are there to help make this decision. Keep in mind that the status report never reports about money actually having been transferred to a merchant, so it is not a complete guarantee that a payment has been finished in that sense. Some tips on using the totals to determine a level of confidence:

Safe route: The safest route to check whether all payments were made is for the merchants to refer to the “Total captured” amount to see whether this equals the “Total registered amount”. While this may be the safest indicator, the downside is that it can sometimes take a long time for acquirers or shoppers to actually have the money transferred and it can be captured.

Quick route: Another option is to see whether the sum of “total shopper pending”, “total acquirer pending” and “total acquirer authorized” matches the “total registered sum”. This implies that everyone responsible has indicated that they are going to make the payment and that the merchant is trusting that everyone will indeed make this. While this route will be faster, it does also have the risk that some payments will actually not have been made.

Balanced route: Depending on the merchant's situation, it can be a good option to only refer to certain totals. For instance, if the merchant only makes use of credit card payments it could be a good route to only look at “Total acquirer approved”, since this will be rather safe but quicker than looking at the captures.

If the merchant does not want to rely on the supplied totals, they can of course also define their decision making on the actual authorization, capture and refund data which is also supplied in the status report and look at the payment method used.

7.5 Possible States and Context States

PaymentState	PaymentContextState	Explanation
NEW	NEW	Merchant has redirected shopper to menu
NEW	RISK_CHECK_OK	Risk score is less than 100 points**
NEW	RISK_CHECK_FAILED	Risk score is at least 100 points**
NEW	REDIRECTED_FOR_AUTHENTICATION	the route might dictate that authentication is required (e.g. 3ds credit cards or paypal).
NEW	AUTHENTICATED	Authentication was not required or has been processed successfully.
NEW	AUTHENTICATION_FAILED	Shopper failed to authenticate him or herself.
NEW	AUTHENTICATION_ERROR	An error has occurred during the authentication process.
NEW	STARTED	intermediate state
STARTED	REDIRECTED_FOR_AUTHORIZATION	Some payment methods require a redirect of the shopper to an issuer site to perform the actual authorization (e.g. IDEAL).
STARTED	AUTHORIZATION_ERROR	An authorization error has occurred.
STARTED	AUTHORIZATION_FAILED	Shopper failed to the authorization process or the issuer declined the authorization request.
AUTHORIZED/PAID*	AUTHORIZED	Payment has been authorized by the financial institution (please take not of the accompanying confidence level).
AUTHORIZED/PAID*	capture=new	A capture request has been created.
AUTHORIZED/PAID*	capture=started	Capture has been batched (prior to sending the request to the financial institution).
AUTHORIZED/PAID*	capture=cancelled	Merchant has cancelled the request.

AUTHORIZED/PAID*	capture=failed	Financial Instituation could not capture the authorized payment.
AUTHORIZED/PAID*	capture=error	An error has occurred during the capture process.
PAID	capture=captured	The authorized payment has been captured succesfully.
CANCELLED	CANCELLED	Payment has been refused by the financial institution, the risk check processor or has been cancelled by the merchant.
CHARGED-BACK	n/a	A charge-baqck has been processed by the financial institution.
CONFIRMED_PAID	n/a	Docdata Payments has received the funds.
CONFIRMED_CHARGEDBACK	n/a	The financial institution has received the charge-back.
CLOSED_SUCCESS	n/a	The payment has been transferred to the merchant.
CLOSED_CANCELLED	n/a	The paymentOrder has been cancelled and has expired.

* depending on merchant setting

** only applicable if merchant uses Docdata Payments fraud module.

In the Order API status report the payment context state is communicated (`<status>AUTHORIZED</status>`), whereas the Command API uses the payment status. At present the Order API status report, communicates up until the capture and refund of authorized payments.