wirecard

Payment Processing API

Merchant Specification

Version: 1.13

This document has been created by Wirecard AG. Its contents may be changed without prior notice. External web links are provided for information only. Wirecard does not claim liability for access to and correctness of the referenced content.

COPYRIGHT

The information contained in this document is intended only for the person or entity to which it is addressed and contains confidential and/or privileged material. Any review, retransmission, dissemination or other use of, or taking of any action in reliance upon, this information by persons or entities other than the intended recipient is prohibited. If you received this in error, please contact Wirecard AG and delete the material from any computer.

Printed in Germany / European Union

Version 1.13

Last Updated: March 2015

TRADEMARKS

The Wirecard logo is a registered trademark of Wirecard AG. Other trademarks and service marks in this document are the sole property of the Wirecard AG or their respective owners.

CONTACT INFORMATION

For questions relating to this document please contact:

Wirecard AG
Einsteinring 35
D-85609 Dornach / Aschheim
Germany

Phone: +49 89 4424 1640
Email: support@wirecard.com

Content

1 Introduction

1.1 Audience

This specification is intended to be read by the technical staff in the merchant's organization responsible for implementing Wirecard's Payment Gateway via Wirecard's Payment Processing API. It is assumed that the reader has a working knowledge of the programming languages required to integrate.

Document Conventions:

- □ The `monospace`font is used for example code and code listings, file names, commands, path names, directory names, Hypertext Markup Language (HTML) tags, and any text that must be typed on the screen.
- □ The *italic* font is used in code to represent placeholder parameters (variables) that should be replaced with an actual value, or items that require emphasis.

The document describes how to create or access resources, provides examples of communications, as well as a series of appendices providing terminology and tables of needed data.

1.2 Revision History

This specification is periodically updated to document design changes made to the Wirecard Payment Gateway. With each revision a new entry is added to the table below, including the date of and the reason for the version change.

Date	Version	Comments
2011-06-10	1.0.0	First Release
2011-08-29	1.1.1	Updated glossary, transaction statuses and transaction types
2012-02-24	1.1.2	Updated glossary, transaction statuses and transaction types.
2012-02-27	1.1.3	Formatting changes and added xml element to glossary
2012-04-19	1.1.4	Added card types
2012-04-24	1.1.5	PaReq case standard, fixed mistakes
2012-05-15	1.1.6	General Review, added tokenize transaction type.
2012-06-19	1.1.7	Changes for 3D Secure codes, new glossary and api redirect items.
2012-06-27	1.1.8	Changes for PayPal, iDEAL, and sofortbanking.
2012-07-11	1.1.9	Added detokenize transaction type.
2012-06-29	1.2.0	Changes for Direktüberweisung
2012-08-28 2012-10-02 2012-10-04 2012-10-10 2012-10-12	1.2.1	Added new payment method Sofortüberweisung Updated the field descriptor for PM sofortbanking from optional in mandatory. Added the maximum length for <firstname> and <lastname> for sofortbanking. Added the maximum length for descriptor for sofortbanking. Updated the field Order Number for PM sofortbanking from optional in mandatory.
2013-04-13 2013-04-17	1.2.2	Update of the field bank-code, bank-account, IBAN and BIC for sofortbanking. Update of the fields Account Number and Bank Code for Direktüberweisung. Deleting the fields IBAN and BIC for Direktüberweisung.

2013-04-22 2013-04-24 2013-05-02	1.2.3	Added the new payment method Skrill Wallet (Draft). Updated the field ‘locale’, ‘requested-amount’ and ‘gross-amount’ for the payment method Skrill Wallet. Updated the fields account-holder.email and wallet-account-id of Skrill Wallet (chapter 6.2.1).These fields are M/O dependent on which of both fields is filled. Updated the description of Skrill Wallet (chapter 6.2). Added the Flexible Split Gateway Option.
2013-06-17 2013-07-03 2013-07-11 2013-08-19	1.2.4	Added the new Skrill functionality Skrill Credits. Changed the max. length for the field descriptor for Direktüberweisung. Deleted the SEPA payment method because there will be a new SEPA implementation. Added the notification timeframe for Direktüberweisung and Sofortüberweisung. Correction of typo ‘PAReq’ into ‘PaReq’ and ‘PARres” into ‘PaRres’ for 3-D-Secure.
2013-10-01	1.3	Addition of section 7 – Direct Debit and Credit Transfer Addition of sections 7.1, 7.2, and 7.3 Update of Appendix E: Transaction Codes Addition of Appendix J
2013-11-22	1.3.1	Updating the notfication description in chapter 5 Adding the notificationURL parameter for Direktüberweisung (5.2.1), Sofortüberweisung (5.3.1) and Skrill Wallet (6.2.1).
2013-12-12	1.4	Updating section 7.3 with new test merchant data Updating section 7.3.1 – Direct Debit Recurring functionality Updating section 7.3.1.2 with XML samples for Direct Debit Recurring Clarification of the due date calculation Clarification of mandatory fields for Direct Debit vs. Credit Transfer
2014-03-27	1.5	Added section 7.4.1 SEPA Direct Debit Authorization Payments Added section 7.6.1 SEPA Void Functionality Added section 7.7.1 Deposits Added section 7.8.1 Bank Data Validations Appendix A: Added Payment Method Wire Transfer Appendix B: Added Transaction Type Deposit Appendix E: Updated with new transaction statuses

2014-04-04	1.5.1	New section 6.4 PayPal added. New section 6.5 ePlatby added. New section 7.8.1.2 BIC Validations added Appendix A updated with ePlatby. Appendix B updated with authorization-only transaction. Appendix B updated with PayPal and ePlatby transactions.
2014-05-21	1.5.2	Minor text changes in Section 7.2 SEPA Appendix B updated transaction type per payment method table
2014-06-02	1.5.3	Section 3.3 and 3.4: changed “Request Methods” from POST to GET.
2014-06-19	1.6	Section 7.3.1: Removed that recurring functionality only available at the end of January 2014 Section 7.3.1: Added Section “Referencing to a “ Section 7.4.1: Added Section “Referencing to a “FIRST” Recurring Transaction” Appendix E: Updated list of transaction statuses
2014-07-14	1.6.1	Section 5.3.1: Added regular expression for IBAN and BIC Section 7.2.1: Added regular expression for IBAN and BIC and Creditor ID Section 7.4.1: Added regular expression for IBAN and BIC Section 7.5.1: Added regular expression for IBAN and BIC Section 3.2.5: Added NVP samples for Credit Card Payments
2014-09-18	1.7	Section 6.6: RatePAY added Section 7.2.1: Updated section 7.2.1.1 with a special due date calculation for Germany Appendix A updated with RatePAY. Appendix B updated with RatePAY transactions. Appendix E updated with new codes
2014-09-25	1.7.1	Section 6.6: RatePAY updated Appendix B: transaction types matrix transponded
2014-10-14	1.8	Section 7.8.1.3: IBAN Only added
2014-10-22	1.9	Section 6.7 MasterPass added Appendix A updated with MasterPass. Appendix B updated with MasterPass transactions.
2014-11-19	1.10	Fixed typo: replaced Descriptor with descriptor Section 4.0: Extended 3-D Secure documentation to cover Amex SafeKey Section 6.4.8 Granting 3rd party permission on PayPal added. Section 6.4 PayPal updated. Section 7.6.1: Added that the cut-off time for Void currently includes weekends
2014-12-10	1.11	Section 6.6: RatePAY Invoice and Direct Debit were renamed to Guaranteed Invoice, Direct Debit and Installments.
2014-12-11	1.11	Section 6.8 Carrier Billing added

2015-01-15	1.12	Section 7.6.1: Update text under first bullet point under ‘Restriction’ Section 7.2.1.1: Update link to Bundesbank under ‘SEPA Direct Debit Specific Fields’ Section 7.4.1.3: New chapter ‘XML – Sample for SEPA Authentication and Pending Direct Debit Request’ Section 8 Appendix A: Enhanced description for Credit Card and EFT Section 13 Appendix F: Update Link for Currency Several wording enhancements and reformatting
2015-03-11	1.13	Section 8.4.5 Cross merchant account references added to the PayPal chapter.
2015-03-24	1.13	Chapter 2 Integration Options added. Chapter 3 Payment Page added.

If there are any errors found, please email your Wirecard Representative, with a detailed description of the error including document version number, page number, and section number.

2 Integration Options

2.1 Overview

There are two different integration paths available for Elastic Engine payment gateway.

2.1.3 Payment Page

Payment Page stands for secure, flexible, easy and transparent payment processing. It provides all payment methods in ONE solution, from credit cards, 3D secure, and direct debits to online payments, mobile payments, alternate payment methods and industry-specific solutions.

Payment Page takes care of payment method selection, collecting credit card details or redirecting consumer to alternative payment provider's site. Merchant is exposed only to limited PCI scope (self-questionnaire A) with this solution.

Payment Page comes in two flavors: Hosted Payment Page where buyer is redirected to Wirecard's server to finish the checkout procedure; and Embedded Payment page where payment screen is displayed in overlay directly on merchant's checkout page.

Only HTML knowledge is required to integrate.

2.1.4 REST API

The maximum control solution, integration to the shop is completely up to the merchant. Requests are constructed by merchant and are sent over HTTP protocol to Elastic Engine end-point where a REST API is exposed. Integration is possible in two ways: XML requests sent from merchant's backend (server-to-server) or JSON requests sent directly from the browser (client-to-server).

Working programmer knowledge is required to integrate this way.

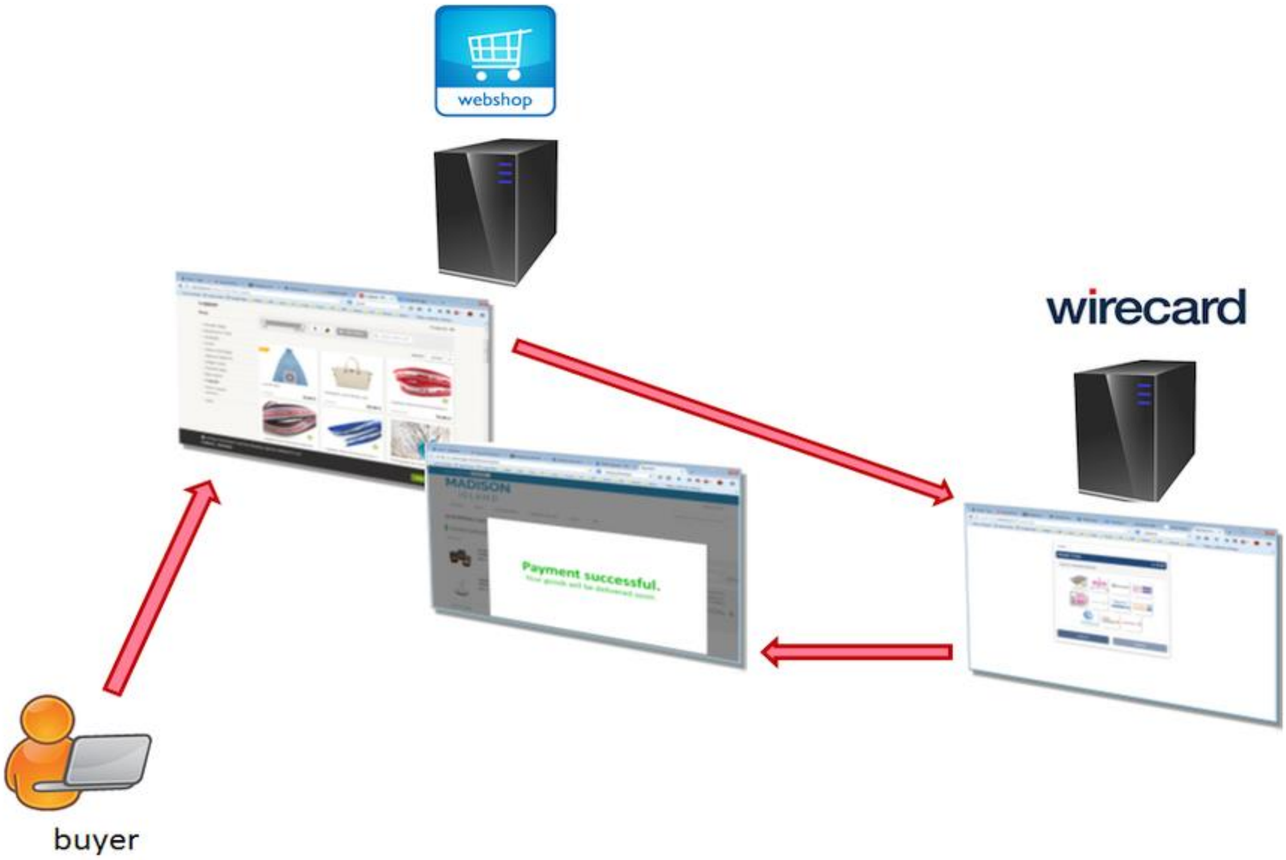
NOTE: The integration via Payment Page is described in chapter 3 Payment Page, the rest of this document referrers to implementation using REST API, though e.g. description of payment methods is common for both options.

3 Payment Page

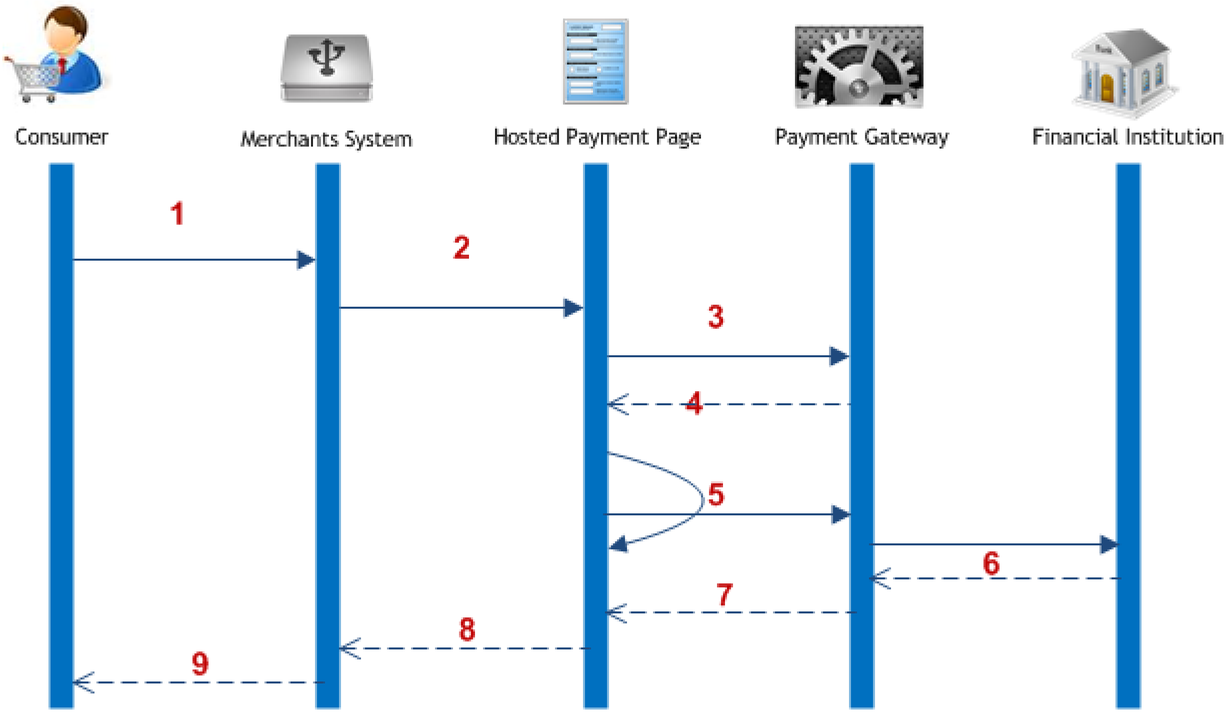
Provides bit more than just an endpoint, it could take care of payment method selection, collecting credit card details or redirecting to alternative payment provider's site. Payment Page comes in two flavors: Hosted Payment Page and Embedded Payment page.

3.1 Hosted Payment Page

Hosted Payment Page integration involves the merchant redirecting the consumer to the Payment Page along with information about the transaction and a secure digital signature. The URL for the Hosted Payment Page varies depending on the installation domain i.e. https://www.somedomain.com/engine/hpp/. Please contact your support representative for the exact URL.



Hosted Payment Page provides a generic workflow available for various payment methods.



1. Consumer checks out with the purchased goods / services.
2. Merchant system redirects to the Hosted Payment Page URL with a digitally signed payment request.
3. The Payment Gateway validates the signature and returns the list of Payment Methods to which the merchant has access with the associated redirect URL.
4. The Hosted Payment Page displays the payment methods from which the consumer may choose.
   Please note: this payment method selection may optionally be skipped based on the merchant's account configuration.
   Depending on the chosen payment method and the respective specifics of this payment method, the Hosted Payment Page then displays the relevant input fields.
5. The Payment Page posts the payment transaction to the payment gateway.
6. The payment gateway processes the payment with the respective financial institution.
7. The payment gateway returns the final response to the Payment Page, which is a digitally signed response message.
8. The Payment Page posts the digitally signed response back to the merchant's success or failure URL.
9. Optionally, the merchant's system validates the signature, and decodes the response message.
   Finally, the consumer (the consumer's browser) is redirected to the merchant's success or failure shop page.

Integration is as easy as inserting an HTML form with hidden fields into merchant’s checkout page, only visible component is the button. The HTML form must POST to correct HPP endpoint. The minimum sample for redirecting to Hosted Payment Page with payment method selection screen is below.

```
<html>
<head>
  <meta charset="utf-8"/>
  <title>HPP Shopping Cart</title>
</head>
<body>
  <form id="hppform" method="POST" role="form" enctype="application/x-www-form-
urlencoded" accept-charset="UTF-8" action="http://test-
```

```
api.wirecard.com/engine/hpp/">
    <input id="wirecard_pay_btn" type="submit" value="Pay"/>
    <input type="hidden" name="merchant_account_id" value="61e8c484-dbb3-4b695"/>
    <input type="hidden" name="request_id" value="c68b9039-968d-1c6b-d9f6-545d"/>
    <input type="hidden" name="request_time_stamp" value="20150226084718"/>
    <input type="hidden" name="payment_method" value=""/>
    <input type="hidden" name="transaction_type" value="auto-sale"/>
    <input type="hidden" name="requested_amount" value="2.56"/>
    <input type="hidden" name="requested_amount_currency" value="EUR"/>
    <input type="hidden" name="locale" value="en"/>
    <input type="hidden" name="request_signature" value="ghty6e4c4730486d180cca"/>
  </form>
</body>
</html>
```

The minimum sample for redirecting to Hosted Payment Page with credit card form directly is below (please note that just fields “payment_method” and “transaction_type” differ).

```
<html>
<head>
  <meta charset="utf-8"/>
  <title>HPP Shopping Cart</title>
</head>
<body>
  <form id="hppform" method="POST" role="form" enctype="application/x-www-form-
urlencoded" accept-charset="UTF-8" action="http://test-
api.wirecard.com/engine/hpp/">
    <input id="wirecard_pay_btn" type="submit" value="Pay with card"/>
    <input type="hidden" name="merchant_account_id" value="61e8c484-dbb3-4b695"/>
    <input type="hidden" name="request_id" value="c68b9039-968d-1c6b-d9f6-545d"/>
    <input type="hidden" name="request_time_stamp" value="20150226084718"/>
    <input type="hidden" name="payment_method" value="creditcard"/>
    <input type="hidden" name="transaction_type" value="authorization"/>
    <input type="hidden" name="requested_amount" value="2.56"/>
    <input type="hidden" name="requested_amount_currency" value="EUR"/>
    <input type="hidden" name="locale" value="en"/>
    <input type="hidden" name="request_signature" value="7fa46e4c4730486d180cca"/>
  </form>
</body>
</html>
```

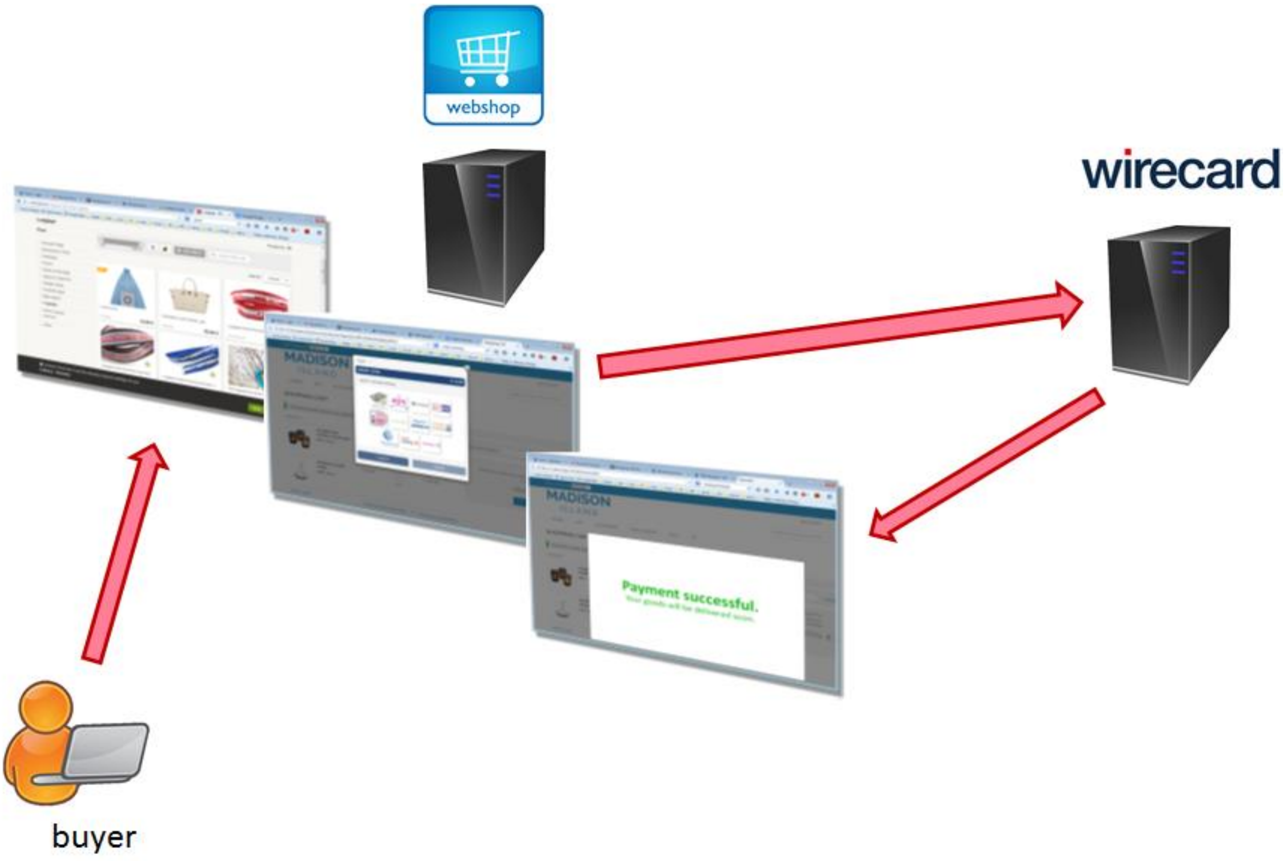
The sample for button redirecting to PayPal checkout portal is below (please note that just fields “payment_method” and “transaction_type” differ).

```
<html>
<head>
  <meta charset="utf-8"/>
  <title>HPP Shopping Cart</title>
</head>
<body>
  <form id="hppform" method="POST" role="form" enctype="application/x-www-form-
urlencoded" accept-charset="UTF-8" action="http://test-
api.wirecard.com/engine/hpp/">
    <input id="wirecard_pay_btn" type="submit" value="Pay with PayPal"/>
    <input type="hidden" name="merchant_account_id" value="61e8c484-dbb3-4b695"/>
    <input type="hidden" name="request_id" value="c68b9039-968d-1c6b-d9f6-545d"/>
    <input type="hidden" name="request_time_stamp" value="20150226084718"/>
    <input type="hidden" name="payment_method" value="paypal"/>
    <input type="hidden" name="transaction_type" value="debit"/>
    <input type="hidden" name="requested_amount" value="2.56"/>
    <input type="hidden" name="requested_amount_currency" value="EUR"/>
    <input type="hidden" name="locale" value="en"/>
    <input type="hidden" name="request_signature" value="25896e4c4730486d180cca"/>
  </form>
</body>
</html>
```

3.2 Embedded Payment Page

With this flavor of integration, Payment Page is displayed in overlay directly on merchant's checkout page. No redirection is needed to select a payment method or to enter credit card details, therefore it makes for much better consumer's experience, who can see that it is still integral part of merchant's site.

Embedded Payment Page integration involves the function call of (along with information about the transaction and a secure digital signature) of the provided JavaScript library. The URL for library varies depending on the installation domain i.e. https://www.somedomain.com/engine/hpp/embeddedLoader.js. Please contact your support representative for the exact URL.



From technical point of view, the integration is even easier than with traditional solution – merchant just includes one JavaScript library

```
<script src="http://api-test.wirecard.com/engine/hpp/embeddedLoader.js"
type="text/javascript"></script>
```

and a function call of that library provides the payment functionality.

```
WirecardHPP.pay(requestData);
```

The minimum sample for showing Embedded Payment Page with payment method selection screen is below.

```
<html>
<head>
  <meta charset="utf-8"/>
  <title>EPP Shopping Cart</title>
  <script src="http://api-test.wirecard.com/engine/hpp/embeddedLoader.js"
type="text/javascript"></script>
</head>
<body>
  <input id="wd_pay_btn" type="button" onclick="pay()" value="Pay"/>
  <script type="text/javascript">
  function pay() {
    var requestData = {
      merchant_account_id: "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
      request_id: "c68b9039-968d-1c6b-d9f6-27e9ab2bcb3e",
      request_time_stamp: “20150226084718”
      payment_method: "",
      transaction_type: "auto-sale",
      requested_amount: "2.56",
      requested_amount_currency: "EUR",
      locale: "en",
      request_signature: "7fa46e4c4730486d180cca590bc2e8dea2a2f912bd175395636a37"
    }
    WirecardHPP.pay(requestData);
  }
  </script>
</body>
</html>
```

The minimum sample showing Embedded Payment Page with credit card form directly is below (please note that just fields “payment_method” and “transaction_type” differ).

```
<html>
<head>
  <meta charset="utf-8"/>
  <title>EPP Shopping Cart</title>
  <script src="http://api-test.wirecard.com/engine/hpp/embeddedLoader.js"
type="text/javascript"></script>
</head>
<body>
  <input id="wd_pay_btn" type="button" onclick="pay()" value="Pay with card"/>
  <script type="text/javascript">
  function pay() {
    var requestData = {
      merchant_account_id: "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
      request_id: "c68b9039-968d-1c6b-d9f6-27e9ab2bcb3e",
      request_time_stamp: “20150226084718”
      payment_method: "creditcard",
      transaction_type: "authorization",
      requested_amount: "2.56",
      requested_amount_currency: "EUR",
      locale: "en",
      request_signature: "7fa46e4c4730486d180cca590bc2e8dea2a2f912bd175395636a37"
    }
    WirecardHPP.pay(requestData);
  }
  </script>
</body>
</html>
```

3.3 Consumer Redirects

Upon a successful or failed Payment Page transaction, the consumer is redirected (via automated HTTP POST) back to the successful or failed URL along with the digitally signed payment message (please refer to 3.5.4 Response Signature). It is strongly recommended merchants verify the signature and/or utilize IPN to determine a transaction's final status.

Default Success & Failure URLs are configured during merchant account setup. To adjust these please contact technical support.

3.4 Fields

Most of the fields from REST API are available also for Hosted Payment Page and Embedded Payment Page, the difference being the usage of underscore instead of hyphen. The following table describes the fields that may appear in the HTML form in the case of HPP or as data for JavaScript call in case of EPP.

3.4.3 Request Fields

Field	Cardinality	Datatype	Size	Description
request_time_stamp	Mandatory	yyyyMMddHHmmss	14	The UTC time-stamp that represents the request.
request_signature	Mandatory	Alphanumeric	64	Refer to SHA-256 Request Signature
merchant_account_id	Mandatory	Alphanumeric	36	A unique identifier assigned for every Merchant Account.
request_id	Mandatory	Alphanumeric	64	The unique string that the Merchant sends with every Transaction in order to uniquely identify it. The Merchant System can subsequently request the status or existence of a Transaction using this identifier.
entry_mode	Optional	Alphanumeric	n/a	The method that the Account Holder information was collected. Possible values include 'empty' (unknown source), 'ecommerce' (collected over the Internet), 'mail-order' (collected over mail order), 'telephone-order' (collected over telephone),

				and 'pos' (collected by the primary payment instrument).
periodic_type	Optional	Alphanumeric	n/a	Indicates how and why a Payment occurs more than once. Possible values include 'installment': one in a set that completes a financial transaction and 'recurring': one in a set that occurs repeatedly, such as a subscription.
sequence_type	Optional	Alphanumeric	n/a	Used in conjunction with periodic_type to indicate the sequence. Possible values include 'first': first transaction in a set, 'recurring': subsequent transactions in the set, 'final': the last transaction in the set.
transaction_type	Mandatory	Alphanumeric	30	A unique identifier assigned for every Transaction Type.
requested_amount	Mandatory	Numeric	18,2	The only amount that accompanies the Transaction when it is created and/or requested. In the case of a Sale or Refund, this is what the Merchant requests. In the case of a Chargeback, this is the amount that is being contested.
requested_amount_currency	Mandatory	Alphanumeric	3	The currency in which a transaction is originally completed.
first_name	Optional	Alphanumeric	32	The first name of the Account Holder.
last_name	Mandatory	Alphanumeric	32	The last name of the Account Holder.
token_id	Optional - Or Card Number	Numeric	36	A unique identifier assigned for every Card Token. This is a surrogate value for the Primary Account Number.
card_type	Mandatory - Or Card Token	Alphanumeric	15	A card scheme accepted by the Processing System. This includes physically issued cards.
account_number	Mandatory - Or Card Token	Numeric	36	The embossed or encoded number that identifies the card issuer to which a transaction is to be routed and the account to which it is to be charged unless specific instructions indicate

				otherwise. In the case of a Credit Card, this is the Primary Account Number.
expiration_month	Mandatory - Or Card Token	Numeric	2	The 2-digit representation of the expiration month of the Card Account.
expiration_year	Mandatory - Or Card Token	Numeric	4	The 4-digit representation of the expiration year of the Card Account.
card_security_code	Optional - Depending on merchant account settings	Numeric	4	A security feature for credit or debit card transactions, providing increased protection against credit card fraud. The Card Security Code is located on the back of MasterCard, Visa and Discover credit or debit cards and is typically a separate group of 3 digits to the right of the signature strip. On American Express cards, the Card Security Code is a printed, not embossed, group of four digits on the front towards the right.
redirect_url	Optional	Alphanumeric	256	The URL where the Account Holder will be redirected to following Transaction completion.
ip_address	Optional	Alphanumeric	15	The IP Address of the Cardholder as recorded by the entity receiving the Transaction Attempt from the Cardholder.
email	Optional	Alphanumeric	64	The email address of the Account Holder.
phone	Optional	Alphanumeric	32	The phone number of the Account Holder.
order_detail	Optional	Alphanumeric	256	Merchant-provided string to store the 'Order Detail' for the Transaction.
order_number	Optional	Alphanumeric	64	Merchant-provided string to store the 'Order Detail' for the Transaction.
merchant_crm_id	Optional	Alphanumeric	64	The Merchant CRM Id for the Account Holder.
field_name_*n[1-3]*	Optional	Alphanumeric	36	Text used to name the Transaction Custom Field Field.
field_value_*n[1-3]*	Optional	Alphanumeric	256	Used with a key, the content used to define the value of the Transaction Custom Field Field.

notification_url_*n[1-3]*	Optional	Alphanumeric	256	The URL used to notify of the Transaction Notification.
notification_transaction_state_*n[1-3]*	Optional	Alphanumeric	12	The Transaction State that triggers the Transaction Notification.
descriptor	Optional	Alphanumeric	64	The field, which is shown on the customer's card statement. This feature is not supported by all the acquirers. The size of this field depends on the acquirer. Please contact technical support for further clarification.
parent_transaction_id	Optional	Alphanumeric	36	A unique identifier assigned for every Parent Transaction.
payment_method	Optional	Alphanumeric	15	Text used to name the Payment Method.
locale	Optional	Alphanumeric	6	Code to indicate which default language the payment page should be rendered in. Refer to Locales
device_fingerprint	Optional	Alphanumeric	4096	A device fingerprint is information collected about a remote computing device for the purpose of identification. Fingerprints can be used to fully or partially identify individual users or devices even when cookies are turned off.

3.4.4 Response Fields

Field	Cardinality	Datatype	Size	Description
response_signature	Mandatory	Alphanumeric	64	Refer to SHA-256 Response Signature
transaction_type	Mandatory	Alphanumeric	30	A unique identifier assigned for every Transaction Type.
transaction_state	Mandatory	Alphanumeric	12	The current status of a Transaction. Typically, a Transaction will start from a Submitted State, to an In-Progress, and then finish in either the Success or Failed state.
transaction_id	Mandatory	Alphanumeric	36	A unique identifier assigned for every Transaction.
request_id	Mandatory	Alphanumeric	64	The unique string that the Merchant sends with every Transaction in order to uniquely

				identify it. The Merchant System can subsequently request the status or existence of a Transaction using this identifier.
requested_amount	Mandatory	Numeric	18,2	The only amount that accompanies the Transaction when it is created and/or requested. In the case of a Sale or Refund, this is what the Merchant requests. In the case of a Chargeback, this is the amount that is being contested.
merchant_account_id	Mandatory	Alphanumeric	36	A unique identifier assigned for every Merchant Account.
completion_time_stamp	Mandatory	yyyyMMddHHmmss	14	The UTC time-stamp that represents the response.
status_code_n	Mandatory	Alphanumeric	12	The status of a Transaction. This is primarily used in conjunction with the Transaction State to determine the exact details of the status of the Transaction.
status_description_n	Mandatory	Alphanumeric	256	Text used to describe the Transaction Status.
status_severity_n	Mandatory	Alphanumeric	20	The severity of the Transaction, can be information, warning, error.
provider_transaction_id_n	Optional	Alphanumeric	36	The unique identifier for a Provider Transaction, typically generated by the Provider.
provider_transaction_reference_id	Optional	Alphanumeric		Provider's reference ID. This may be non-unique.
authorization_code	Mandatory	Alphanumeric	36	An alphanumeric numeric code used to represent the Provider Transaction Provider Authorization.
token_id	Optional - Or Card Number	Alphanumeric	36	A unique identifier assigned for every Card Token. This is a surrogate value for the Primary Account Number.
masked_account_number	Optional	Alphanumeric	36	A code used to represent the Card Masked Account.
ip_address	Optional	Alphanumeric	15	The IP Address of the Cardholder as recorded by the entity receiving the Transaction Attempt from the Cardholder.

3.5 Security

As payment information is exchanged between the merchant's system, the consumer's browser, and the payment gateway, it is important that the data exchange safeguards against man-in-middle attacks.

Both Hosted Payment Page and Embedded Payment Page use a digital SHA-256 signature for all message exchanges. The signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit.

3.5.3 Request Signature

When the Merchant creates the client side HTML form or JavaScript data, the following values are to be concatenated, leading and trailing space removed, and SHA-256 applied to the combined string

1. request_time_stamp
2. request_id
3. merchant_account_id
4. transaction_type
5. requested_amount
6. requested_amount_currency
7. redirect_url
8. ip_address
9. [Secret Key] (provided by your support representative)

The SHA-256 hash value is then presented on the Merchants client side form as “request_signature“ field. Please note the order of the fields is important. Also, note the same values used in the Request Signature must be placed into the client side form/data (with the exception of the Secret Key).

An example of Request Signature generation is as follows:

```
request_time_stamp          = '20120430123012'
request_id                  = 'order-12345'
merchant_account_id         = 'b19fb056-d8da-449b-ac85-cfbfd0558914'
transaction_type            = 'purchase'
requested_amount            = '1.01'
requested_amount_currency   = 'USD'
redirect_url                = ''
ip_address                  = '127.0.0.1'
secret_key                  = 'efabf47b-e43b-4785-873f-1c5bc65b7cd2'

Pre SHA-256 string
20120430123012order-12345b19fb056-d8da-449b-ac85-
cfbfd0558914purchase1.01USD127.0.0.1efabf47b-e43b-4785-873f-1c5bc65b7cd2

SHA-256 signature
e93ed221efb8f6048df31794609d9557f65f175659e4928d10463b8998e3f61f
```

3.5.4 Response Signature

Within the response message the following values are concatenated, leading and trailing space removed, and SHA-256 applied to the combined string:

1. merchant_account_id
2. transaction_id
3. request_id
4. transaction_type
5. transaction_state

6. completion_time_stamp
7. token_id
8. masked_account_number
9. ip_address
10. authorization_code
11. [Secret Key] (provided by your support representative)

3.5.5 Signature Generation Code Samples

PHP

```
$request_signature = hash('sha256', trim($request_time_stamp . $request_id .
$merchant_account-id . $transaction_type . $requested_amount .
$request_amount_currency . $redirect_url . $ip_address . $secret_key));
```

C# / ASP.NET

```
public static string GetSHA256(string text) {
       byte[] hashValue;
       byte[] message = Encoding.UTF8.GetBytes(text);

       SHA256Managed hashString = new SHA256Managed();
       string hex = "";

       hashValue = hashString.ComputeHash(message);
       foreach( byte x in hashValue)
       {
              hex += String.Format("{0:x2}", x);
       }
       return hex.Trim();
}
```

Java

```
private static String tosha256(String... fields) {
    StringBuffer sb = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        sb = new StringBuffer();
        for (String field : fields) {
            sb.append(field.trim());
        }
        md.update(sb.toString().getBytes("utf-8"));
        byte[] mdbytes = md.digest();
        return DatatypeConverter.printHexBinary(mdbytes);
    } catch (NoSuchAlgorithmException e) {
        sb = null;
    } catch (UnsupportedEncodingException e) {
        sb = null;
    }
    return sb == null ? null : sb.toString();
}
```

Groovy

```
import java.security.MessageDigest;
...
```

```
def messageDigest = MessageDigest.getInstance("SHA256");
def secret_key = 'XXXXXXXXXXXXXXXXXXXXXX';

def   stringToHash   =   time_stamp   +   request_id   +   merchant_account_id   +
transaction_type + requested_amount + requested_amount_currency + redirect_url +
ip_address + secret_key;

messageDigest.update( stringToHash.trim().getBytes() );
def shaHex = new BigInteger(1, messageDigest.digest()).toString(16);
```

3.6 Payment Selection

Payment Page displays payment methos selection screen if the field “payment_method” is left out empty and merchant has configured more than one payment method in the payment gateway. The list of shown payment methods might be narrowed by specifying “transaction_type” field, value “auto-sale” might be used to display all available payment methods.

Payment Page supports automatic selection of payment methods in order to bypass rendering of the payment method selection screen. Simply set “payment_method” in the initial payment request to force selection of a specific payment method.

Payment Page supports prepopulation of consumer information to reduce or completely eliminate consumer data entry. Consumer information such as name, order, and address information can be sent within the initial payment request. Payment Page will use these values as the default, which can be manually overriden by the consumer.

When processing credit card transactions via Payment Page there is the option to completely bypass rendering of the Payment Page. Subsequent processing redirects such as 3D secure are automatically handled. To enable SilentPay processing the merchant must pass the following fields within the initial payment request: account_number, card_type, card_security_code, expiration_year, expiration_month.

3.7 Configuration

Payment methods, transaction types, card types, and supported currencies are configured within the payment gateway and are common for REST API and Payment Page. Please contact your support representative to obtain the Secret Key for signing Payment Page requests.

4 Accessing Resources

4.1 Overview

The Payment Processing API conforms to the design principles of Representational State Transfer (REST). Wirecard's REST APIs provide access to resources (data entities) via URI paths. To use a REST API, your application will make an HTTP request and parse the response. By default, the response format is XML. If you wish, you can request JSON instead of XML. Your methods will be the standard HTTP methods such as GET, PUT, POST and DELETE.

The API currently supports the following data formats: for requests XML or NVP, and for responses XML, HTML, JSON or NVP with some methods only accepting a subset of these formats. Simply change the format extension to a request or set the HTTP Accept header to get results in the format of your choice.

Methods to retrieve data from the Payment Processing API require a `GET` request. Methods that submit data require a `POST`. API methods that require a particular HTTP method will return an error if you do not make your request using the correct method. HTTP Response Codes are meaningful.

Some API methods take optional or requisite parameters. Two things to keep in mind when making requests with parameters:

- □ Parameter values should be UTF-8
- □ All for REST URLs should be URL encoded.

Because the Payment Processing API is based on open standards, you can use any web development language to access the API.

4.2 Security

Every request to Wirecard's Payment Gateway is authorized and authenticated. Users are authenticated using basic authentication over SSL enabled channel. The Merchant Account configuration is used to authorize the request.

4.3 Card Security – Tokenization

Every card number that accompanies a transaction in Wirecard's Payment Gateway is subsequently tokenized. Regardless of the outcome of the transaction, this token can be subsequently used instead of the clear card account number, in any subsequent transaction. This means that the client system never needs to store the sensitive card information, helping to reduce PCI DSS Compliance issues.

It is also possible to submit a Tokenization Transaction, whereby the transaction does not process a payment. It simply tokenizes information. The Detokenize Transaction can be utilized to subsequently retrieve this tokenized information.

5 Submitting Transactions

5.1 Overview

The Payment Processing API methods allow merchants to submit payment transactions as well as search for payments and merchant data. In addition, tokenization transactions are permitted.

5.2 Create a Payment

This attempts to create a transaction. The type of the transaction is determined by the payload of the message.

URL	`https://hostname/engine/rest/payments/`
Request Formats	XML, NVP
Response Formats	XML, HTML, JSON
Request Methods	POST

5.2.1 Fields

The following table describes the fields that appear in the payment resource.

Term	Man /Opt	Format	Max Size
account-holder.address.city	O	Alphanumeric	32
account-holder.address.country	O	Alphanumeric	3
account-holder.address.postal-code	O	Alphanumeric	16
account-holder.address.state	O	Alphanumeric	32
account-holder.address.street1	O	Alphanumeric	128
account-holder.address.street2	O	Alphanumeric	128
account-holder.date-of-birth	O	Date	0
account-holder.email	O	Alphanumeric	64
account-holder.first-name	M	Alphanumeric	32
account-holder.gender	O	Alphanumeric	1
account-holder.last-name	M	Alphanumeric	32
account-holder.merchant-crm-id	O	Alphanumeric	64
account-holder.phone	O	Alphanumeric	32
account-holder.social-security-number	O	Alphanumeric	14
bank-account.account-number	O	Alphanumeric	34
bank-account.bank-code	O	Alphanumeric	15
bank-account.bic	M - for iDEAL	Alphanumeric	5
bank-account.iban	M - for iDEAL	Alphanumeric	4
cancel-redirect-url	O	Alphanumeric	256

card.account-number	M - Or Card Token	Alphanumeric	36
card.card-security-code	Depending on merchant account settings	Alphanumeric	4
card.card-type	M - for creditcard.	Alphanumeric	15
card.expiration-month	M - Or Card Token	Numeric	2
card.expiration-year	M - Or Card Token	Numeric	4
card.token-id	O - Or Card Number	Alphanumeric	36
card-token.masked-account-number	O	Alphanumeric	36
custom-field.field-name	O	Alphanumeric	36
custom-field.field-value	O	Alphanumeric	256
descriptor	O	Alphanumeric	64
fail-redirect-url	O	Alphanumeric	256
ip-address	O	Alphanumeric	15
merchant-account-id	M	Alphanumeric	36
notification.transaction-state	O	Alphanumeric	12
order-detail	O	Alphanumeric	65535
order-number	O	Alphanumeric	64
parent-transaction-id	O	Alphanumeric	36
payment-methods.payment-method-name	O	Alphanumeric	15
processing-redirect-url	O	Alphanumeric	256
redirect-url	O	Alphanumeric	56
requested-amount	M	Numeric	18,2
request-id	M	Alphanumeric	64
shipping.address.city	O	Alphanumeric	32
shipping.address.country	O	Alphanumeric	3
shipping.address.postal-code	O	Alphanumeric	16
shipping.address.state	O	Alphanumeric	32
shipping.address.street1	O	Alphanumeric	128
shipping.address.street2	O	Alphanumeric	128
shipping.first-name	O	Alphanumeric	32
shipping.last-name	O	Alphanumeric	32
shipping.phone	O	Alphanumeric	32
status.code	M	Alphanumeric	12
status.description	M	Alphanumeric	256
status.severity	M	Alphanumeric	20
statuses.status	M	Alphanumeric	12
success-redirect-url	O	Alphanumeric	256
three-d.acs-url	O	Alphanumeric	256
three-d.cardholder-authentication-status	O	Alphanumeric	16000

three-d.cardholder-authentication-value	O	Alphanumeric	1024
three-d.eci	O	Numeric	2
three-d.pareq	O	Alphanumeric	16000
three-d.pares	O	Alphanumeric	16000
three-d.xid	O	Alphanumeric	36
transaction-id	M	Alphanumeric	36
transaction-type	M	Alphanumeric	30

5.2.2 Storing Additional Data with Custom Fields

In addition to processing transactions, Wirecard’s Payment Gateway also permits the storage and later retrieval of additional information. The use of ‘Custom Fields’ permits the client application to store key-value pairs with each transaction.

In the following example, the client application has stored a purchase order, invoice, crm-id, customer tier, and promotional code. This information is also echoed back in any response querying the status of a transaction. It is also possible to see this information in the reporting system.

```
<custom-fields>
   <custom-field field-name="purchase-order" field-value="PO999888776655111"/>

   <custom-field field-name="crm-id" field-value="7766558" />
   <custom-field field-name="customer-tier" field-value="gold" />
   <custom-field field-name="promotional-code" field-value="2010-06-15" />
</custom-fields>
```

5.2.3 Instant Payment Notifications

Wirecard’s Platform also has a built-in notification capability. The Merchant simply sends the notifications instructions as part of each transaction request. Notifications need to be specified as a URL. There are two types of notifications:

- □ **HTTP(S)** (Web Server POST)
- □ **SMTP** (Email prefixed with ‘mailto’)

It is possible to setup conditional notifications based on the state of the transaction. For example, a Merchant can instruct a notification to only occur on ‘failed’ or ‘successful’ transactions.

In the following example, the transaction will notify the Merchant’s web site for a failed transaction, and send all results to an email address.

```
<?xml version="1.0" encoding="UTF-8"?>
<notifications>
   <notification transaction-state="failed" url="https://www.merchant.com/ipn" />
   <notification url="mailto:admin@merchant.com" />
</notifications>
```

5.2.4 Sample XML Request and Response – Credit Card

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
      <merchant-account-id>c3671cf9-c775-4e39-8d67-31ce24094682</merchant-
account-id>
      <request-id>e942228e-0697-4993-94f8-b418ad27c0a9</request-id>
      <transaction-type>purchase</transaction-type>
      <requested-amount currency="USD">1.01</requested-amount>
```

```
        <account-holder>
                <first-name>John</first-name>
                <last-name>Doe</last-name>
                <email>john.doe@test.com</email>
                <phone></phone>
                <address>
                        <street1>123 anystreet</street1>
                        <city>Brantford</city>
                        <state>ON</state>
                        <country>CA</country>
                </address>
        </account-holder>
        <card>
                <account-number>4444333322221111</account-number>
                <expiration-month>12</expiration-month>
                <expiration-year>2020</expiration-year>
                <card-type>visa</card-type>
                <card-security-code>123</card-security-code>
        </card>
        <ip-address>127.0.0.1</ip-address>
</payment>
```

Sample Response:

```
<?xml version="1.0" standalone="yes"?>
<payment    self="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-
8d67-31ce24094682/payments/a236ad78-a66d-11e1-b76c-005056ab0016"
xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id
ref="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-8d67-
31ce24094682">c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
        <transaction-id>a236ad78-a66d-11e1-b76c-005056ab0016</transaction-id>
        <request-id>e942228e-0697-4993-94f8-b418ad27c0a9</request-id>
        <transaction-type>purchase</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2012-05-25T13:29:20.169Z</completion-time-stamp>
        <statuses>
                <status  code="201.0000"  description="3d-acquirer:The  resource  was
successfully created." severity="information"/>
        </statuses>
        <requested-amount currency="USD">1.01</requested-amount>
        <card-token>
                <token-id>4894643804401111</token-id>
                <masked-account-number>444433******1111</masked-account-number>
        </card-token>
        <ip-address>127.0.0.1</ip-address>
        <descriptor>graypay Amazon</descriptor>
        <authorization-code>153620</authorization-code>
</payment>
```

5.2.5 Sample NVP Request

NVP is a canonical form of the XML Representation. The following is a sample. Field names are derived from the XML Element Names by using an underscore delimiter. All fields should be URL-encoded.

```
POST http://10.0.0.142:8080/engine/rest/payments/ HTTP/1.1
```

```
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.0.0.142:8080
Content-Length: 526

first_name=John&card_type=visa&transaction_type=authorization&expiration_month=12&accou
nt_number=4444333322221111&expiration_year=2013&card_security_code=123&payment_ip_addre
ss=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=123
anystreet&address_street2=Suite%20999&address_city=Toronto&address_state=ON&address_cou
ntry=CA&requested_amount_currency=USD&request_id=e05c2d16-a4e6-4794-9ac0-
93b15a1aac99&merchant_account_id=c3671cf9-c775-4e39-8d67-
31ce24094682&address_postal_code=M4P1E8&requested_amount=1.01
```

3.2.5.1 With headers "Content-Type application/x-www- form-urlencoded; charset=UTF-8" and "accept application/xml"

Sample Request:

```
POST https://i-elastic-stack0.wirecard.sys:443/engine/rest/payments/ HTTP/1.1
Connection: close
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=
Accept: application/xml
Content-Length: 528
Host: i-elastic-stack0.wirecard.sys:443
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

first_name=WDCP&card_type=visa&transaction_type=authorization&expiration_month=12&accou
nt_number=4444333322221111&expiration_year=2020&card_security_code=123&payment_ip_addre
ss=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=Einsteinring
35&address_street2=Suite%20999&address_city=Toronto&address_state=ON&address_country=CA
&requested_amount_currency=USD&request_id=c5a1c782-0191-4753-a88d-
2460336644b4&merchant_account_id=33531c0f-81ec-4527-a087-
8d395bf8f365&address_postal_code=M4P1E8&requested_amount=1.01
```

Sample Response:

```
HTTP/1.1 201 Created
Date: Mon, 30 Jun 2014 11:11:30 GMT
Location:    https://i-elastic-stack0.wirecard.sys/engine/rest/merchants/33531c0f-
81ec-4527-a087-8d395bf8f365/payments/498c3ee6-0047-11e4-b97d-00163e000156
Content-Type: application/xml;charset=UTF-8
Content-Language: en-US
Content-Length: 1542
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment" self="http://i-
elastic-stack0.wirecard.sys:9000/engine/rest/merchants/33531c0f-81ec-4527-a087-
8d395bf8f365/payments/498c3ee6-0047-11e4-b97d-00163e000156">
    <merchant-account-id ref="http://i-elastic-
stack0.wirecard.sys:9000/engine/rest/config/merchants/33531c0f-81ec-4527-a087-
8d395bf8f365">33531c0f-81ec-4527-a087-8d395bf8f365</merchant-account-id>
    <transaction-id>498c3ee6-0047-11e4-b97d-00163e000156</transaction-id>
    <request-id>c5a1c782-0191-4753-a88d-2460336644b4</request-id>
    <transaction-type>authorization</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2014-06-30T11:11:30.000Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="3d-acquirer:The resource was
successfully created." severity="information" provider-transaction-
id="C837002140412669055889"/>
    </statuses>
    <avs-code>U</avs-code>
    <csc-code>P</csc-code>
    <requested-amount currency="USD">1.01</requested-amount>
    <card-token>
        <token-id>4851746269541111</token-id>
        <masked-account-number>444433******1111</masked-account-number>
    </card-token>
    <descriptor>demo descriptor</descriptor>
    <notifications>
        <notification url="https://i-elastic-
stack0.wirecard.sys:9443/merchant/rest/success"/>
    </notifications>
    <authorization-code>846556</authorization-code>
    <api-id>elastic-api</api-id>
</payment>
```

3.2.5.2 with headers "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" and "accept:application/json"

Sample Request:

```
POST https://i-elastic-stack0.wirecard.sys:443/engine/rest/payments/ HTTP/1.1

Connection: close

Content-Type: application/x-www-form-urlencoded;charset=UTF-8

Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=

Accept: application/json

Content-Length: 528

Host: i-elastic-stack0.wirecard.sys:443

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)


first_name=WDCP&card_type=visa&transaction_type=authorization&expiration_month=12&accou
nt_number=4444333322221111&expiration_year=2020&card_security_code=123&payment_ip_addre
ss=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=Einsteinring
35&address_street2=Suite%20999&address_city=Toronto&address_state=ON&address_country=CA
&requested_amount_currency=USD&request_id=b5747371-4839-4d0c-a3b2-
e7f17624d258&merchant_account_id=33531c0f-81ec-4527-a087-
```

```
8d395bf8f365&address_postal_code=M4P1E8&requested_amount=1.01
```

Sample Response:

```
HTTP/1.1 201 Created
Date: Mon, 30 Jun 2014 11:17:44 GMT
Location:     https://i-elastic-stack0.wirecard.sys/engine/rest/merchants/33531c0f-
81ec-4527-a087-8d395bf8f365/payments/2888f832-0048-11e4-b97d-00163e000156
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/json;charset=UTF-8
Content-Language: en-US
Connection: close
Transfer-Encoding: chunked

{
   "cruiseNameValuePairBean":    {
      "passengerName": null,
      "lodgingCheckInDate": null,
      "lodgingCheckOutDate": null,
      "lodgingRoomRate": null,
      "numberOfNights": null,
      "lodgingName": null,
      "lodgingCityName": null,
      "lodgingRegionCode": null,
      "lodgingCountryCode": null,
      "agentCode": null,
      "ticketNumber": null,
      "carrierCode": null,
      "itinerarySegmentFlightNumber5": null,
      "itinerarySegmentFlightNumber6": null,
      "itinerarySegmentFlightNumber7": null,
      "itinerarySegmentFlightNumber8": null,
      "itinerarySegmentFlightNumber9": null,
      "itinerarySegmentFlightNumber10": null,
      "itinerarySegmentFareClass1": null,
      "itinerarySegmentFareClass2": null,
      "itinerarySegmentFareClass3": null,
      "itinerarySegmentFareClass4": null,
      "itinerarySegmentFareClass5": null,
      "itinerarySegmentFareClass6": null,
      "itinerarySegmentFareClass7": null,
      "itinerarySegmentFareClass8": null,
      "itinerarySegmentFareClass9": null,
      "itinerarySegmentFareClass10": null,
      "itinerarySegmentFareBasis1": null,
      "itinerarySegmentFareBasis2": null,
      "itinerarySegmentFareBasis3": null,
      "itinerarySegmentFareBasis4": null,
      "itinerarySegmentFareBasis5": null,
      "itinerarySegmentFareBasis6": null,
      "itinerarySegmentFareBasis7": null,
      "itinerarySegmentFareBasis8": null,
      "itinerarySegmentFareBasis9": null,
      "itinerarySegmentFareBasis10": null,
      "itinerarySegmentStopOverCode1": null,
      "itinerarySegmentStopOverCode2": null,
      "itinerarySegmentStopOverCode3": null,
      "itinerarySegmentStopOverCode4": null,
      "itinerarySegmentStopOverCode5": null,
      "itinerarySegmentStopOverCode6": null,
      "itinerarySegmentStopOverCode7": null,
      "itinerarySegmentStopOverCode8": null,
      "itinerarySegmentStopOverCode9": null,
      "itinerarySegmentStopOverCode10": null,
      "itinerarySegmentTaxAmount1": null,
      "itinerarySegmentTaxAmount2": null,
      "itinerarySegmentTaxAmount3": null,
      "itinerarySegmentTaxAmount4": null,
      "itinerarySegmentTaxAmount5": null,
      "itinerarySegmentTaxAmount6": null,
      "itinerarySegmentTaxAmount7": null,
      "itinerarySegmentTaxAmount8": null,
      "itinerarySegmentTaxAmount9": null,
      "itinerarySegmentTaxAmount10": null,
      "itinerarySegmentCarrierCode1": null,
      "itinerarySegmentCarrierCode2": null,
      "itinerarySegmentCarrierCode3": null,
      "itinerarySegmentCarrierCode4": null,
      "itinerarySegmentCarrierCode5": null,
      "itinerarySegmentCarrierCode6": null,
      "itinerarySegmentCarrierCode7": null,
      "itinerarySegmentCarrierCode8": null,
      "itinerarySegmentCarrierCode9": null,
      "itinerarySegmentCarrierCode10": null,
      "itinerarySegmentDepartureAirportCode1": null,
      "itinerarySegmentDepartureAirportCode2": null,
      "itinerarySegmentDepartureAirportCode3": null,
      "itinerarySegmentDepartureAirportCode4": null,
      "itinerarySegmentDepartureAirportCode5": null,
      "itinerarySegmentDepartureAirportCode6": null,
      "itinerarySegmentDepartureAirportCode7": null,
      "itinerarySegmentDepartureAirportCode8": null,
      "itinerarySegmentDepartureAirportCode9": null,
      "itinerarySegmentDepartureAirportCode10": null,
      "itinerarySegmentDepartureCityCode1": null,
      "itinerarySegmentDepartureCityCode2": null,
      "itinerarySegmentDepartureCityCode3": null,
      "itinerarySegmentDepartureCityCode4": null,
      "itinerarySegmentDepartureCityCode5": null,
      "itinerarySegmentDepartureCityCode6": null,
      "itinerarySegmentDepartureCityCode7": null,
      "itinerarySegmentDepartureCityCode8": null,
      "itinerarySegmentDepartureCityCode9": null,
      "itinerarySegmentDepartureCityCode10": null,
      "itinerarySegmentArrivalAirportCode1": null,
      "itinerarySegmentArrivalAirportCode2": null,
      "itinerarySegmentArrivalAirportCode3": null,
      "itinerarySegmentArrivalAirportCode4": null,
      "itinerarySegmentArrivalAirportCode5": null,
      "itinerarySegmentArrivalAirportCode6": null,
      "itinerarySegmentArrivalAirportCode7": null,
      "itinerarySegmentArrivalAirportCode8": null,
      "itinerarySegmentArrivalAirportCode9": null,
      "itinerarySegmentArrivalAirportCode10": null,
      "itinerarySegmentArrivalCityCode1": null,
      "itinerarySegmentArrivalCityCode2": null,
    "itinerarySegmentArrivalCityCode3": null,
    "itinerarySegmentArrivalCityCode4": null,
    "itinerarySegmentArrivalCityCode5": null,
    "itinerarySegmentArrivalCityCode6": null,
    "itinerarySegmentArrivalCityCode7": null,
    "itinerarySegmentArrivalCityCode8": null,
    "itinerarySegmentArrivalCityCode9": null,
    "itinerarySegmentArrivalCityCode10": null,
    "itinerarySegmentDepartureDate1": null,
    "itinerarySegmentDepartureDate2": null,
    "itinerarySegmentDepartureDate3": null,
    "itinerarySegmentDepartureDate4": null,
    "itinerarySegmentDepartureDate5": null,
    "itinerarySegmentDepartureDate6": null,
    "itinerarySegmentDepartureDate7": null,
    "itinerarySegmentDepartureDate8": null,
    "itinerarySegmentDepartureDate9": null,
    "itinerarySegmentDepartureDate10": null,
    "itinerarySegmentArrivalDate1": null,
    "itinerarySegmentArrivalDate2": null,
    "itinerarySegmentArrivalDate3": null,
    "itinerarySegmentArrivalDate4": null,
    "itinerarySegmentArrivalDate5": null,
    "itinerarySegmentArrivalDate6": null,
    "itinerarySegmentArrivalDate7": null,
    "itinerarySegmentArrivalDate8": null,
    "itinerarySegmentArrivalDate9": null,
    "itinerarySegmentArrivalDate10": null,
    "itinerarySegmentFlightNumber1": null,
    "itinerarySegmentFlightNumber2": null,
    "itinerarySegmentFlightNumber3": null,
    "itinerarySegmentFlightNumber4": null,
    "packageTypeCode": null,
    "emptyCruiseData": true
 },
 "payment":    {
    "statuses": {"status": [      {
       "value": null,
       "code": "201.0000",
       "description": "3d-acquirer:The resource was successfully created.",
       "severity": "information",
       "provider-transaction-id": "C811656140412706466621"
    }]},
    "shipping": null,
    "card": null,
    "descriptor": "demo descriptor",
    "notifications": {"notification": [      {
       "value": null,
       "transaction-state": null,
       "url": "https://i-elastic-stack0.wirecard.sys:9443/merchant/rest/success"
    }]},
    "mandate": null,
    "locale": null,
    "files": null,
    "invoice": null,
    "periodic": null,
    "wallet": null,
    "country": null,
    "settlement": null,
    "batch": null,
    "consumer": null,
    "device": null,
    "pos": null,
    "iso": null,
    "merchant-account-id":       {
       "value": "33531c0f-81ec-4527-a087-8d395bf8f365",
       "ref":                                                 "http://i-elastic-
stack0.wirecard.sys:9000/engine/rest/config/merchants/33531c0f-81ec-4527-a087-
8d395bf8f365"
    },
    "transaction-id": "2888f832-0048-11e4-b97d-00163e000156",
    "request-id": "b5747371-4839-4d0c-a3b2-e7f17624d258",
    "transaction-type": "authorization",
    "transaction-state": "success",
    "completion-time-stamp": 1404127064000,
    "avs-code": "U",
    "csc-code": "P",
    "requested-amount":       {
       "value": 1.01,
       "currency": "USD"
    },
    "card-token":       {
       "token-id": "4851746269541111",
       "masked-account-number": "444433******1111"
    },
    "authorization-code": "931277",
    "api-id": "elastic-api",
    "self":                                                   "http://i-elastic-
stack0.wirecard.sys:9000/engine/rest/merchants/33531c0f-81ec-4527-a087-
8d395bf8f365/payments/2888f832-0048-11e4-b97d-00163e000156"
 },
 "nameValuePairBean":    {
    "amount": "1.01",
    "requestId": "b5747371-4839-4d0c-a3b2-e7f17624d258",
    "transactionType": "authorization",
    "city": null,
    "postalCode": null,
    "street1": null,
    "ipAddress": null,
    "orderNumber": null,
    "orderDetail": null,
    "orderItems": [],
    "street2": null,
    "firstName": "WDCP",
    "lastName": "Doe",
    "email": "test@test.com",
    "dateOfBirth": null,
    "phone": "",
    "merchantCrmId": null,
    "socialSecurityNumber": null,
    "tokenId": null,
    "maskedAccountNumber": null,
    "consumerId": null,
    "redirectUrl": null,
    "processingRedirectUrl": null,
    "cancelRedirectUrl": null,
    "failRedirectUrl": null,
    "successRedirectUrl": null,
    "dueDate": null,
    "entryMode": null,
    "riskReferenceId": null,
    "accountNumber": "4444333322221111",
    "expirationMonth": "12",
    "expirationYear": "2020",
    "cardSecurityCode": "123",
    "cardType": "visa",
    "signedDate": null,
    "signedCity": null,
    "bankName": null,
    "bankCode": null,
    "iban": null,
    "bic": null,
    "signatureImage": null,
    "periodicType": null,
    "sequenceType": null,
    "recordNumber": null,
    "mandateId": null,
    "paymentMerchantAccountResolverCategory": null,
    "paymentParentTransactionId": null,
    "apiid": null,
    "paymentMethodName": null,
    "devicePolicyScore": null,
    "deviceFingerprint": null,
    "bankAccountOwner": null,
    "customFieldName1": null,
    "customFieldValue1": null,
    "customFieldName2": null,
    "customFieldValue2": null,
    "notificationTransactionState3": null,
    "paymentMerchantAccountId": "33531c0f-81ec-4527-a087-8d395bf8f365",
    "requestTimeStamp": null,
    "requestSignature": null,
    "attempt3d": null,
    "bankAccountNumber": null,
    "customFieldName3": null,
    "customFieldValue3": null,
    "notificationTransactionUrl": null,
    "notificationTransactionState": null,
    "loyaltyCardAccountNumber": null,
    "loyaltyCardPin": null,
    "loyaltyCardPromotionCode": null,
    "loyaltyCardActivityDate": null,
    "notificationUrl1": null,
    "notificationTransactionState1": null,
    "notificationUrl2": null,
    "notificationTransactionState2": null,
    "notificationUrl3": null,
    "currency": "USD",
    "descriptor": null,
    "state": null,
    "fileName": null,
    "country": null,
    "locale": null
 },
 "airlineNameValuePairBean":    {
    "passengerName": null,
    "airlineCode": null,
    "airlineName": null,
    "passengerCode": null,
    "passengerPhone": null,
    "passengerEmail": null,
    "agentCode": null,
    "ticketNumber": null,
    "passengerIpAddress": null,
    "ticketIssueDate": null,
    "ticketRestrictedFlag": null,
    "pnrFileKey": null,
    "ticketCheckDigit": null,
    "agentName": null,
    "nonTaxableNetAmount": null,
    "numberOfPassengers": null,
    "reservationCode": null,
    "itinerarySegmentFlightNumber5": null,
    "itinerarySegmentFlightNumber6": null,
    "itinerarySegmentFlightNumber7": null,
    "itinerarySegmentFlightNumber8": null,
    "itinerarySegmentFlightNumber9": null,
    "itinerarySegmentFlightNumber10": null,
    "itinerarySegmentFareClass1": null,
    "itinerarySegmentFareClass2": null,
    "itinerarySegmentFareClass3": null,
    "itinerarySegmentFareClass4": null,
    "itinerarySegmentFareClass5": null,
    "itinerarySegmentFareClass6": null,
    "itinerarySegmentFareClass7": null,
    "itinerarySegmentFareClass8": null,
    "itinerarySegmentFareClass9": null,
    "itinerarySegmentFareClass10": null,
    "itinerarySegmentFareBasis1": null,
    "itinerarySegmentFareBasis2": null,
    "itinerarySegmentFareBasis3": null,
    "itinerarySegmentFareBasis4": null,
    "itinerarySegmentFareBasis5": null,
    "itinerarySegmentFareBasis6": null,
    "itinerarySegmentFareBasis7": null,
    "itinerarySegmentFareBasis8": null,
    "itinerarySegmentFareBasis9": null,
    "itinerarySegmentFareBasis10": null,
    "itinerarySegmentStopOverCode1": null,
    "itinerarySegmentStopOverCode2": null,
    "itinerarySegmentStopOverCode3": null,
    "itinerarySegmentStopOverCode4": null,
    "itinerarySegmentStopOverCode5": null,
    "itinerarySegmentStopOverCode6": null,
    "itinerarySegmentStopOverCode7": null,
    "itinerarySegmentStopOverCode8": null,
    "itinerarySegmentStopOverCode9": null,
    "itinerarySegmentStopOverCode10": null,
    "itinerarySegmentTaxAmount1": null,
    "itinerarySegmentTaxAmount2": null,
    "itinerarySegmentTaxAmount3": null,
    "itinerarySegmentTaxAmount4": null,
    "itinerarySegmentTaxAmount5": null,
    "itinerarySegmentTaxAmount6": null,
    "itinerarySegmentTaxAmount7": null,
    "itinerarySegmentTaxAmount8": null,
    "itinerarySegmentTaxAmount9": null,
    "itinerarySegmentTaxAmount10": null,
    "ticketIssuerStreet1": null,
    "ticketIssuerStreet2": null,
    "ticketIssuerCity": null,
    "ticketIssuerState": null,
    "ticketIssuerCountry": null,
    "ticketIssuerPostalCode": null,
    "itinerarySegmentCarrierCode1": null,
    "itinerarySegmentCarrierCode2": null,
    "itinerarySegmentCarrierCode3": null,
    "itinerarySegmentCarrierCode4": null,
    "itinerarySegmentCarrierCode5": null,
    "itinerarySegmentCarrierCode6": null,
    "itinerarySegmentCarrierCode7": null,
    "itinerarySegmentCarrierCode8": null,
    "itinerarySegmentCarrierCode9": null,
    "itinerarySegmentCarrierCode10": null,
    "itinerarySegmentDepartureAirportCode1": null,
    "itinerarySegmentDepartureAirportCode2": null,
    "itinerarySegmentDepartureAirportCode3": null,
    "itinerarySegmentDepartureAirportCode4": null,
    "itinerarySegmentDepartureAirportCode5": null,
    "itinerarySegmentDepartureAirportCode6": null,
    "itinerarySegmentDepartureAirportCode7": null,
    "itinerarySegmentDepartureAirportCode8": null,
    "itinerarySegmentDepartureAirportCode9": null,
    "itinerarySegmentDepartureAirportCode10": null,
    "itinerarySegmentDepartureCityCode1": null,
    "itinerarySegmentDepartureCityCode2": null,
    "itinerarySegmentDepartureCityCode3": null,
    "itinerarySegmentDepartureCityCode4": null,
    "itinerarySegmentDepartureCityCode5": null,
    "itinerarySegmentDepartureCityCode6": null,
    "itinerarySegmentDepartureCityCode7": null,
    "itinerarySegmentDepartureCityCode8": null,
    "itinerarySegmentDepartureCityCode9": null,
    "itinerarySegmentDepartureCityCode10": null,
    "itinerarySegmentArrivalAirportCode1": null,
    "itinerarySegmentArrivalAirportCode2": null,
    "itinerarySegmentArrivalAirportCode3": null,
    "itinerarySegmentArrivalAirportCode4": null,
    "itinerarySegmentArrivalAirportCode5": null,
    "itinerarySegmentArrivalAirportCode6": null,
    "itinerarySegmentArrivalAirportCode7": null,
    "itinerarySegmentArrivalAirportCode8": null,
    "itinerarySegmentArrivalAirportCode9": null,
    "itinerarySegmentArrivalAirportCode10": null,
    "itinerarySegmentArrivalCityCode1": null,
    "itinerarySegmentArrivalCityCode2": null,
    "itinerarySegmentArrivalCityCode3": null,
    "itinerarySegmentArrivalCityCode4": null,
    "itinerarySegmentArrivalCityCode5": null,
    "itinerarySegmentArrivalCityCode6": null,
    "itinerarySegmentArrivalCityCode7": null,
    "itinerarySegmentArrivalCityCode8": null,
    "itinerarySegmentArrivalCityCode9": null,
    "itinerarySegmentArrivalCityCode10": null,
    "itinerarySegmentDepartureDate1": null,
    "itinerarySegmentDepartureDate2": null,
    "itinerarySegmentDepartureDate3": null,
    "itinerarySegmentDepartureDate4": null,
    "itinerarySegmentDepartureDate5": null,
    "itinerarySegmentDepartureDate6": null,
      "itinerarySegmentDepartureDate7": null,
      "itinerarySegmentDepartureDate8": null,
      "itinerarySegmentDepartureDate9": null,
      "itinerarySegmentDepartureDate10": null,
      "itinerarySegmentArrivalDate1": null,
      "itinerarySegmentArrivalDate2": null,
      "itinerarySegmentArrivalDate3": null,
      "itinerarySegmentArrivalDate4": null,
      "itinerarySegmentArrivalDate5": null,
      "itinerarySegmentArrivalDate6": null,
      "itinerarySegmentArrivalDate7": null,
      "itinerarySegmentArrivalDate8": null,
      "itinerarySegmentArrivalDate9": null,
      "itinerarySegmentArrivalDate10": null,
      "itinerarySegmentFlightNumber1": null,
      "itinerarySegmentFlightNumber2": null,
      "itinerarySegmentFlightNumber3": null,
      "itinerarySegmentFlightNumber4": null
   }
}
```

3.2.5.3 with headers "Content-Type: application/x-www-form-urlencoded; charset=UTF-8 (no "accept" header was sent)

Sample Request:

```
POST https://i-elastic-stack0.wirecard.sys:443/engine/rest/payments/ HTTP/1.1
Connection: close
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=
Content-Length: 528
Host: i-elastic-stack0.wirecard.sys:443
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

first_name=WDCP&card_type=visa&transaction_type=authorization&expiration_month=12&accou
nt_number=4444333322221111&expiration_year=2020&card_security_code=123&payment_ip_addre
ss=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=Einsteinring
35&address_street2=Suite%20999&address_city=Toronto&address_state=ON&address_country=CA
&requested_amount_currency=USD&request_id=8ecdfca4-3d5b-4574-b428-
2262a56fa111&merchant_account_id=33531c0f-81ec-4527-a087-
8d395bf8f365&address_postal_code=M4P1E8&requested_amount=1.01
```

Sample Response:

```
HTTP/1.1 201 Created
Date: Mon, 30 Jun 2014 11:28:50 GMT
Location: https://i-elastic-stack0.wirecard.sys/engine/rest/merchants/33531c0f-81ec-
4527-a087-8d395bf8f365/payments/b4eb1f8e-0049-11e4-b97d-00163e000156
Content-Type: text/plain;charset=ISO-8859-1
Content-Length: 570
Connection: close

transaction_type=authorization&completion_time_stamp=20140630112849&status_code_1=201.0
```

```
000&status_severity_1=information&token_id=4851746269541111&transaction_state=success&t
ransaction_id=b4eb1f8e-0049-11e4-b97d-00163e000156&merchant_account_id=33531c0f-81ec-
4527-a087-8d395bf8f365&request_id=8ecdfca4-3d5b-4574-b428-
2262a56fa111&requested_amount=1.01&requested_amount_currency=USD&masked_account_number=
444433******1111&status_description_1=3d-
acquirer%3AThe+resource+was+successfully+created.&provider_transaction_id_1=C8278861404
12772968790&authorization_code=801397&
```

3.2.5.4 with headers "Content-Type:application/x-www-form-urlencoded; charset=UTF-8" and "accept: text/html"

Sample Request:

```
POST https://i-elastic-stack0.wirecard.sys:443/engine/rest/payments/ HTTP/1.1
Connection: close
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=
Accept: text/html
Content-Length: 523
Host: i-elastic-stack0.wirecard.sys:443
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

first_name=WDCP&card_type=visa&transaction_type=purchasea&expiration_month=12&acco
unt_number=4444333322221111&expiration_year=2013&card_security_code=123&payment_ip
_address=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=Einste
inring
35&address_street2=Suite%20999&address_city=Munich&address_state=BY&address_countr
y=DE&requested_amount_currency=USD&request_id=31ffd958-994e-4734-ba6a-
4009cd0bb709&merchant_account_id=3e549fd0-50de-11e0-b8af-
0800200c9a66&address_postal_code=M4P1E8&requested_amount=1.01
```

Sample Response:

```
HTTP/1.1 201 Created
Date: Mon, 30 Jun 2014 11:50:50 GMT
Location: https://i-elastic-stack0.wirecard.sys/engine/rest/merchants/3e549fd0-
50de-11e0-b8af-0800200c9a66/payments/c8271870-004c-11e4-b97d-00163e000156
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Set-Cookie: JSESSIONID=76241838CB12567B6825B713241351D5.engine0; Path=/engine/;
HttpOnly
Connection: close
Transfer-Encoding: chunked

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
                <title>Transaction Details</title>
```

```
                 <style type="text/css">
                         table {border-collapse: collapse; font-family: verdana;
font-size: 11px; margin: 0 auto;}
                         th {background-color: #ddd; padding: 5px; font-size:
13px;min-width: 300px;}
                         td {padding: 5px;}
                         td.label {background-color: #eee; font-weight: 900;min-
width: 300px;}
                         td.value {min-width: 100px;}
                         tr.toggle{}
                         div.toggleMe{
                                 width: 800px ;
                                 margin-left: auto;
                                 margin-right: auto;
                                 text-align:left;
                                 color: blue;
                                 font-size:16px;
                                 cursor: pointer;
                                 }
                 </style>
                 <script src="/engine/includes/jquery-
1.7.2.min.js;jsessionid=76241838CB12567B6825B713241351D5.engine0"
type="text/javascript"></script>
                  <script type="text/javascript">
                          $(document).ready(function(){

                                          $("td.value").each(function(){
                                                  if ( $(this).text().trim() ==
""){

        $(this).parent().addClass("toggle");
                                                  }
                                          });

                                          $(".toggle").hide();

                                          $(".toggleMe").click(function() {
                                         $(".toggle").toggle();
                                         //$(this).text("Please click to show
fields");
                                         centerDiv();
                                     }).toggle(
                                                  function(){
                                                          $(this).html("<b>Please
click to SHOW blank fields</b>");
                                                  },
                                                  function(){
                                                          $(this).html("<b>Please
click to HIDE blank fields</b>");
                                                  });

                                          centerDiv();
                                 });
                          function centerDiv(){
                                 // center div
                             var width = $("#mainTable").width();
                             $(".toggleMe").css("width", width+"px");
                          }
                  </script>
        </head>
        <div class="toggleMe"><b>Please click to SHOW blank fields</b></div>
        <body>
                <table id="mainTable" border="1">
                         <tr><th colspan="2"><b>Transaction Details</b></th></tr>
                         <tr><td class="label">Transaction Id</td><td
class="value">c8271870-004c-11e4-b97d-00163e000156</td></tr>
                         <tr><td class="label">Parent Transaction Id</td><td
class="value"></td></tr>
                         <tr><td class="label">Completion Timestamp (GMT)</td><td
class="value">2014-06-30T11:50:50.000Z</td></tr>
                         <tr><td class="label">Merchant Account Id</td><td
class="value">3e549fd0-50de-11e0-b8af-0800200c9a66</td></tr>
                         <tr><td class="label">Transaction Type</td><td
class="value"></td></tr>
                         <tr><td class="label">Transaction State</td><td
class="value">FAILED</td></tr>
                         <tr><td class="label">Requested Amount</td>
                         <td class="value">1.01</td></tr>
                         <tr><td class="label">Currency</td><td
class="value">USD</td></tr>

                         <tr><td class="label">Authorization Code</td><td
class="value"></td></tr>
                         <tr><td class="label">Request Id</td><td
class="value">31ffd958-994e-4734-ba6a-4009cd0bb709</td></tr>
                         <tr><td class="label">Expiration Date</td><td
class="value"></td></tr>
                         <tr><td class="label">Consumer Id</td><td
class="value"></td></tr>
                         <tr><td class="label">IP Address</td><td
class="value"></td></tr>
            <tr><td class="label">Entry Mode</td><td class="value"></td></tr>
            <tr><td class="label">Periodic Type</td><td class="value"></td></tr>
                         <tr><td class="label">Order Number</td><td
class="value"></td></tr>
                         <tr><td class="label">Order Details</td><td
class="value"></td></tr>
                         <tr><td class="label">descriptor</td><td
class="value"></td></tr>
                         <tr><td class="label">API Name</td><td
class="value">elastic-api</td></tr>
                         <tr><td class="label">API Transaction Id</td><td
class="value"></td></tr>
                         <tr><th colspan="2"><b>Card holder Details</b></th></tr>
                         <tr><td class="label">First Name</td><td
```

```
class="value"></td></tr>
                         <tr><td class="label">Last Name</td><td
class="value"></td></tr>
                         <tr><td class="label">Date Of Birth</td><td
class="value"></td></tr>
                         <tr><td class="label">Gender</td><td
class="value"></td></tr>
                         <tr><td class="label">Email</td><td
class="value"></td></tr>
                         <tr><td class="label">Phone</td><td
class="value"></td></tr>
                         <tr><td class="label">Street Line 1</td><td
class="value"></td></tr>
            <tr><td class="label">Street Line 2</td><td class="value"></td></tr>
                         <tr><td class="label">City</td><td
class="value"></td></tr>
            <tr><td class="label">State</td><td class="value"></td></tr>
            <tr><td class="label">Country</td><td class="value"></td></tr>
            <tr><td class="label">Post Code</td><td class="value"></td></tr>
                         <tr><th colspan="2"><b>Card Details</b></th></tr>
                         <tr><td class="label">Card Type</td><td
class="value"></td></tr>
                         <tr><td class="label">Card Token Id</td><td
class="value"></td></tr>
                         <tr><td class="label">Masked Account Number</td><td
class="value"></td></tr>
                         <tr><td class="label">Expiration Month</td><td
class="value"></td></tr>
                         <tr><td class="label">Expiration Year</td><td
class="value"></td></tr>
                         <tr><td class="label">Avs Code</td><td
class="value"></td></tr>
                         <tr><td class="label">Csc Code</td><td
class="value"></td></tr>
                         <tr><th colspan="2"><b>Bank Account</b></th></tr>
                         <tr><td class="label">BIC</td><td
class="value"></td></tr>
                         <tr><td class="label">IBAN</td><td
class="value"></td></tr>
                         <tr><td class="label">Bank Code</td><td
class="value"></td></tr>
                         <tr><td class="label">Account Number</td><td
class="value"></td></tr>
                         <tr><th colspan="2"><b>Device</b></th></tr>
                         <tr><td class="label">Finger Print</td><td
class="value"></td></tr>
                         <tr><td class="label">Policy Score</td><td
class="value"></td></tr>
                         <tr><th colspan="2"><b>Statuses</b></th></tr>

                                  <tr><td class="label">400.1009</td><td>ERROR -
The Transaction Type is invalid. Please check your input and try again.</td></tr>
                                  <tr><td class="label">Provider Transactions
```

```
Id</td><td></td></tr>

                         <tr><th colspan="2"><b>Notifications</b></th></tr>

                         <tr><th colspan="2"><b>Custom Fields</b></th></tr>

            <tr><th colspan="2"><b>Shipping</b></th></tr>
            <tr><td class="label">First Name</td><td class="value"></td></tr>
                         <tr><td class="label">Last Name</td><td
class="value"></td></tr>
            <tr><td class="label">Phone</td><td class="value"></td></tr>
            <tr><td class="label">Street Line 1</td><td class="value"></td></tr>
            <tr><td class="label">Street Line 2</td><td class="value"></td></tr>
            <tr><td class="label">City</td><td class="value"></td></tr>
            <tr><td class="label">State</td><td class="value"></td></tr>
            <tr><td class="label">Country</td><td class="value"></td></tr>
            <tr><td class="label">Post Code</td><td class="value"></td></tr>
            <tr><th colspan="2"><b>Charge Back</b></th></tr>
            <tr><td class="label">Chargeback Id</td><td class="value"></td></tr>
            <tr><td class="label">Chargeback Date</td><td class="value"></td></tr>
            <tr><td class="label">Chargeback Retrieval Date</td><td
class="value"></td></tr>
            <tr><td class="label">Parent Transaction Amount</td><td
class="value"></td></tr>
            <tr><td class="label">Chargeback Conversion Rate</td><td
class="value"></td></tr>
            <tr><td class="label">Chargeback Reason Code</td><td
class="value"></td></tr>
                         <tr><td class="label">Chargeback Provider Transaction
Id</td><td class="value"></td></tr>
            <tr><td class="label">Chargeback Provider Account Id</td><td
class="value"></td></tr>
            <tr><td class="label">Chargeback Additional Info.</td><td
class="value"></td></tr>
            <tr><th colspan="2"><b>Airline</b></th></tr>
            <tr><td class="label">Airline Code</td><td class="value"></td></tr>
            <tr><td class="label">Airline Name</td><td class="value"></td></tr>
            <tr><td class="label">Passenger Code</td><td class="value"></td></tr>
            <tr><td class="label">Passenger Name</td><td class="value"></td></tr>
            <tr><td class="label">Passenger Phone</td><td class="value"></td></tr>
            <tr><td class="label">Passenger Email</td><td class="value"></td></tr>
            <tr><td class="label">Passenger Ip Address</td><td
class="value"></td></tr>
            <tr><td class="label">Ticket Number</td><td class="value"></td></tr>
            <tr><td class="label">Ticket Issue Date</td><td
class="value"></td></tr>
            <tr><td class="label">Ticket Restricted Flag</td><td
class="value"></td></tr>
            <tr><td class="label">Passenger Name File Id</td><td
class="value"></td></tr>
            <tr><td class="label">Ticket Check Digit</td><td
class="value"></td></tr>
            <tr><td class="label">Agent Code</td><td class="value"></td></tr>
            <tr><td class="label">Agent Name</td><td class="value"></td></tr>
            <tr><td class="label">Non-Taxable Net Amount</td><td
class="value"></td></tr>
            <tr><td class="label">Number Of Passengers</td><td
class="value"></td></tr>
            <tr><td class="label">Reservation Code</td><td
class="value"></td></tr>



                 <tr><th colspan="2">&nbsp;</th></tr>
                 </table>
        </body>
</html>
```

5.3 Retrieve Transaction by Transaction Id

Retrieving a transaction using a Transaction Id returns a single transaction belonging to a merchant account. The transaction id attribute must match the value that was included in the transaction response sent by Wirecard. An error is returned if the Transaction Id is not available or the requested user is not authorized to see the content. The desired content type can be set through the Accept header or by specifying an extension.

URL	`https://hostname/engine/rest/merchants/{merchant-account-id}/payments/{transaction-id}`
Request Formats	XML, NVP
Response Formats	XML, HTML, JSON
Request Methods	GET

5.3.1 Fields

Term	Man / Opt	Format	Max Size
merchant-account-id	M	alphanumeric	36
transaction-id	M	alphanumeric	36

5.3.2 Sample Request

```
https://hostname/engine/rest/merchants/ba261be8-af94-11df-ab7800163e5eafd7/payments/048b27e0-9c31-
4cab-9eab-3b72b1b4d498
```

5.3.3 Sample Response

The response corresponds to the Transaction Type of the transaction being looked up. For example, the Transaction Type = ‘purchase’ and the Transaction Type = ‘tokenize’ have different responses when the transaction is being created, and that same response will occur for this retrieval. See the sample responses as described for each Transaction Type.

5.4 Retrieve Transaction by Request Id

This returns a single transaction belonging to a merchant account. The request id attribute must match to a Request Id that was submitted during the creation of a transaction. An error is returned if the request id is not available or the user is not authorized to see the content. The desired content type can be set through the Accept header or by specifying an extension.

URL	`https://hostname/engine/rest/merchants/{merchant-account-id}/payments/search?payment.request-id={request-id}`
Request Formats	XML, NVP
Response Formats	XML, HTML, JSON
Request Methods	GET

5.4.1 Fields

Term	Man / Opt	Format	Max Size
merchant-account-id	M	alphanumeric	36
request-id	M	alphanumeric	150

5.4.2 Sample Request

```
https://hostname/engine/rest/merchants/ba261be8-af94-11df-ab78-
00163e5eafd7/payments/search?payment.request-id=048b27e0-9c31-4cab-9eab-3b72b1b4d498
```

5.4.3 Sample Response

The response corresponds to the Transaction Type of the transaction being looked up. For example, the Transaction Type = ‘purchase’ and the Transaction Type = ‘tokenize’ have different responses when the transaction is being created, and that same response will occur for this retrieval. See the sample responses as described for each Transaction Type.

5.5 Tokenize Credit Card Information

The tokenize Transaction Type simply converts credit card information into a token that can be used in subsequent Payment Transactions, instead of the actual credit card information.

5.5.1 Fields

The fields used in this transaction are a subset of those in a Payment Request, described in section 0.

Term	Man /Opt	Format	Max Size
account-holder.date-of-birth	O	date	0
account-holder.email	O	Alphanumeric	64
account-holder.first-name	O	Alphanumeric	32
account-holder.gender	O	Alphanumeric	1
account-holder.last-name	O	Alphanumeric	32

account-holder.phone	O	Alphanumeric	32
card.account-number	M	Alphanumeric	36
card.card-type	M	Alphanumeric	15
card.expiration-month	M	Numeric	2
card.token-id	O	Alphanumeric	36
card.expiration-year	O	Numeric	4
card-token.masked-account-number	O	Alphanumeric	36
ip-address	O	Alphanumeric	15
merchant-account-id	M	Alphanumeric	36
request-id	M	Alphanumeric	150
status.code	M	Alphanumeric	12
status.description	M	Alphanumeric	256
status.severity	M	Alphanumeric	20
statuses.status	M	Alphanumeric	12
transaction-id	M	Alphanumeric	36
transaction-type	M	Alphanumeric	30

5.5.2 Sample XML Request and Response – Tokenize Credit Card

Sample Request:

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
   <request-id>34ce4f85-f623-4cda-a821-10678db1a845</request-id>
   <transaction-type>tokenize</transaction-type>
   <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <email>john.doe@test.com</email>
      <gender>male</gender>
      <date-of-birth>01-01-1980</date-of-birth>
      <phone>555 555 5555</phone>
      <address>
         <street1>123 anystreet</street1>
         <street2>apt 2</street2>
         <city>Brantford</city>
         <state>ON</state>
         <country>CA</country>
         <postal-code>M4P1E8</postal-code>
      </address>
      <merchant-crm-id>salesforce-76156</merchant-crm-id>
   </account-holder>
   <card>
      <account-number>4444333322221111</account-number>
      <expiration-month>12</expiration-month>
      <expiration-year>2020</expiration-year>
      <card-type>visa</card-type>
   </card>
   <ip-address>127.0.0.1</ip-address>
</payment>
```

Sample Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment"
self="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-8d67-
31ce24094682/payments/a3051862-cb10-11e1-bf4d-005056ab0016">
    <merchant-account-id
ref="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-8d67-
31ce24094682">c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
    <transaction-id>a3051862-cb10-11e1-bf4d-005056ab0016</transaction-id>
    <request-id>34ce4f85-f623-4cda-a821-10678db1a845</request-id>
    <transaction-type>tokenize</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2012-07-11T04:26:51.323Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully
created." severity="information"/>
    </statuses>
    <card-token>
        <token-id>4791271295681111</token-id>
        <masked-account-number>444433******1111</masked-account-number>
    </card-token>
    <ip-address>127.0.0.1</ip-address>
    <api-id>elastic-api</api-id>
</payment>
```

5.6 Detokenize Credit Card Information

The detokenize Transaction Type is the inverse of the tokenize Transaction Type. In that a token-id is provided to retrieve the original information.

5.6.1 Fields

The fields used in this transaction are a subset of those in a Payment Request, described in section 0.

Term	Man /Opt	Format	Max Size
account-holder.date-of-birth	O	date	0
account-holder.email	O	Alphanumeric	64
account-holder.first-name	O	Alphanumeric	32
account-holder.gender	O	Alphanumeric	1
account-holder.last-name	O	Alphanumeric	32
account-holder.phone	O	Alphanumeric	32
card.account-number	M	Alphanumeric	36
card.card-type	M	Alphanumeric	15
card.expiration-month	M	Numeric	2
card.token-id	M	Alphanumeric	36
card.expiration-year	O	Numeric	4
card-token.masked-account-number	O	Alphanumeric	36
ip-address	O	Alphanumeric	15

merchant-account-id	M	Alphanumeric	36
request-id	M	Alphanumeric	150
status.code	M	Alphanumeric	12
status.description	M	Alphanumeric	256
status.severity	M	Alphanumeric	20
statuses.status	M	Alphanumeric	12
transaction-id	M	Alphanumeric	36
transaction-type	M	Alphanumeric	30

5.6.2 Sample XML Request and Response – Detokenize Credit Card

Sample Request:

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
   <request-id>5271295a-3e94-4f92-b67c-3a3754da9d0c</request-id>
   <transaction-type>detokenize</transaction-type>
   <card-token>
      <token-id>4791271295681111</token-id>
   </card-token>
</payment>
```

Sample Response:

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment"
self="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-8d67-
31ce24094682/payments/b5c0e080-cb10-11e1-bf4d-005056ab0016">
    <merchant-account-id
ref="http://10.0.0.142:8080/engine/rest/merchants/c3671cf9-c775-4e39-8d67-
31ce24094682">c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
    <transaction-id>b5c0e080-cb10-11e1-bf4d-005056ab0016</transaction-id>
    <request-id>5271295a-3e94-4f92-b67c-3a3754da9d0c</request-id>
    <transaction-type>detokenize</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2012-07-11T04:27:22.713Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully
created." severity="information"/>
    </statuses>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <email>john.doe@test.com</email>
        <phone>555 555 5555</phone>
        <address>
            <street1>123 anystreet</street1>
            <street2>apt 2</street2>
            <city>Brantford</city>
            <state>ON</state>
            <country>CA</country>
            <postal-code>M4P1E8</postal-code>
        </address>
        <merchant-crm-id>salesforce-76156</merchant-crm-id>
    </account-holder>
    <card>
        <account-number>4444333322221111</account-number>
        <expiration-month>12</expiration-month>
        <expiration-year>2020</expiration-year>
        <card-type>visa</card-type>
    </card>
    <card-token>
        <token-id>4791271295681111</token-id>
    </card-token>
    <api-id>elastic-api</api-id>
</payment>
```

6 3-D Secure

6.1 Overview

3-D Secure is a protocol supported by all major card schemes that provides secure authentication and processing of online payments. Merchants wishing to comply need to integrate the specific 3-D requests and payment parameters.

The Three Domain (3-D) Secure initiative by VISA is a payment standard for secure handling of credit card transactions in electronic commerce. This provides Issuers with the ability to authenticate cardholders during an online purchase. Branded as Verified by Visa and MasterCard SecureCode, Amex SafeKey and JCB J/Secure, 3-D Secure is designed to clearly identify cardholders and accelerate the growth of electronic commerce through increased consumer confidence.

Authentication is accomplished by verification of certain data which is maintained by the card issuing bank and identifies the individual making an online purchase as the legal owner of the card used. 3-D Secure is more than a payment authentication method or a technology definition. It is a model to isolate the liabilities of the various parties involved in the payment transaction cycle. The payment environment requires the participating cardholder to be registered (enrolled) for the process with his issuing bank. In essence, all parties involved in the payment flow must support the 3-D secure transactions. The 3-D framework requires the card issuing and acquiring banks to provide cardholders and merchants with an authentication methodology, without binding them to proprietary technology.

6.2 Merchant Plug-In

A Merchant Plug-In is a software module which provides a communication interface between the merchant and the card schemes' directory servers. It can be integrated in the merchant website or it may be hosted by a service provider or an acquirer. The main functions of an MPI are to verify the card issuers digital signature used in the authentication process, validate enrollment and authentication response messages, encrypt and store passwords and certificates, and retrieve payment records and associated card details to resolve transaction disputes. Merchants can choose to leverage the enrollment check and payment authentication using a hosted or an integrated MPI. Both options will notify the merchant if either of the two validations failed.

Payments need to be submitted with the relevant 3-D Secure artifacts obtained from the MPI. See Section 0 in order to see how to submit the relevant 3-D Secure artifacts. See Appendix G: 3-D-Secure Process Flow to see the complete flow.

6.2.1 Wirecard Merchant Plug-In (MPI)

This document describes the integrated MPI hosted by Wirecard, for merchants that wish to use Wirecard as its 3-D Secure MPI Provider. This solution performs all of the relevant validations and exchanges as required by the card schemes.

6.2.2 3rd Party MPI

If a merchant wishes to use a 3rd Party MPI, then only payments need to be submitted with the relevant 3-D Secure artifacts obtained from the 3rd Party MPI.

6.3 Process Flow

Processing 3-D Secure transactions differs from standard payment processing. Merchants must construct and route their transaction requests accordingly.

The three necessary steps for processing are:

- □ **Check Enrollment** – consists of a single request/response communication that verifies if the card number is eligible and participates in the 3-D Program.
- □ **Customer ACS Communication -** redirects the Customer to the ACS URL, a webpage provided of the cardholder's bank. This allows the Customer to authenticate himself by entering his 3-D credentials.
- □ **Check Payer Response –** Validation of the authentication result. This is used in conjunction with check enrollment.

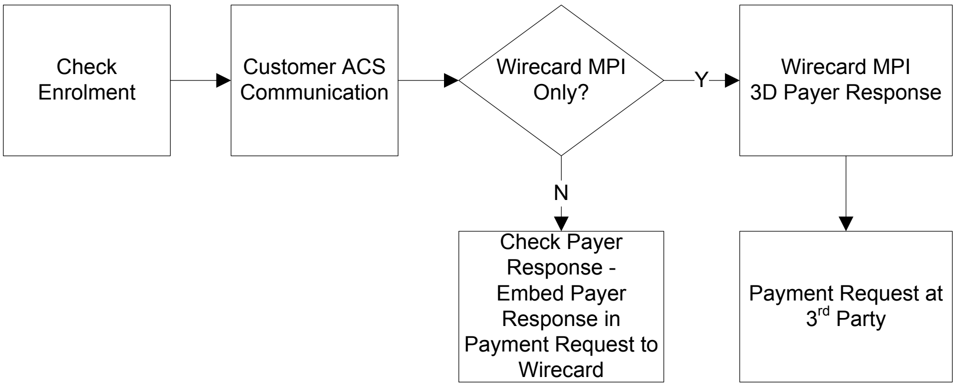

Check Enrolment
Customer ACS Communication
Wirecard MPI Only?
Y
Wirecard MPI 3D Payer Response
N
Check Payer Response - Embed Payer Response in Payment Request to Wirecard
Payment Request at 3rd Party

Figure 1: 3-D Secure Process Flow

See Section 6.4 in order to see how to submit the relevant 3-D Secure artifacts. For a more detailed process flow, refer to Appendix G: 3-D-Secure Process Flow.

6.4 Check Enrollment

Wirecard's Payment Processing API permits merchants to check if a card/cardholder is enrolled in the 3-D Secure program. To verify Enrollment status, the merchant system sends a transaction request to Wirecard's MPI which in turn posts an Enrollment check request (VEReq) to the card schemes' directory servers. In an exchange of messages the directory communicates with the ACS of the card issuer, if required, to determine if the cardholder is enrolled for 3-D Secure. A response message (VERes) is returned by the ACS to the MPI to prove that the cardholder is enrolled or that authentication has been attempted.

6.4.1 Fields

The same Payment Fields are required, with no new input fields, and 2 additional 3-D Secure Output Fields. See Section 0 for a description of the payment fields. The following two response fields are specific to 3-D Secure.

Term	Man / Opt	Format	Max Size
three-d.pareq	M	alphanumeric	16000
three-d.acs-url	M	alphanumeric	100

6.4.2 Sample XML Request and Response

Sample Request:

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
       <merchant-account-id>3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-
account-id>
       <request-id>6eab85bd-9278-4ced-b5cd-1969e78f8bef</request-id>
       <transaction-type>check-enrollment</transaction-type>
       <requested-amount currency="USD">1.00</requested-amount>
       <account-holder>
              <first-name>John</first-name>
              <last-name>Doe</last-name>
              <email>john.doe@test.com</email>
              <phone></phone>
              <address>
                     <street1>123 anystreet</street1>
                     <city>Brantford</city>
                     <state>ON</state>
                     <country>CA</country>
              </address>
       </account-holder>
       <card>
              <account-number>4012000300001003</account-number>
              <expiration-month>01</expiration-month>
              <expiration-year>2019</expiration-year>
              <card-type>visa</card-type>
              <card-security-code>003</card-security-code>
       </card>
       <ip-address>127.0.0.1</ip-address>
</payment>
```

Sample Response:

```
<?xml version="1.0" standalone="yes"?>
<payment     self="http://10.0.0.142:8080/engine/rest/merchants/3e549fd0-50de-11e0-b8af-
0800200c9a66/payments/bb685724-a66d-11e1-b76c-005056ab0016"   xmlns="http://www.elastic-
payments.com/schema/payment">
       <merchant-account-id  ref="http://10.0.0.142:8080/engine/rest/merchants/3e549fd0-
50de-11e0-b8af-0800200c9a66">3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-account-id>
       <transaction-id>bb685724-a66d-11e1-b76c-005056ab0016</transaction-id>
       <request-id>6eab85bd-9278-4ced-b5cd-1969e78f8bef</request-id>
       <transaction-type>check-enrollment</transaction-type>
       <transaction-state>success</transaction-state>
       <completion-time-stamp>2012-05-25T13:30:03.674Z</completion-time-stamp>
       <statuses>
              <status   code="201.0000"   description="3d-acquirer:The   resource   was
successfully created." severity="information"/>
       </statuses>
       <requested-amount currency="USD">1.00</requested-amount>
       <card-token>
              <token-id>4178399322971003</token-id>
              <masked-account-number>401200******1003</masked-account-number>
       </card-token>
       <ip-address>127.0.0.1</ip-address>
       <descriptor>demo descriptor</descriptor>
       <authorization-code/>
```

```
        <three-d>

        <pareq>eJxtk9tOg0AQhu99CsK93RNQ2ixrqrUeEo22VeMlXSYtCScXsK1P77LFQwsQEmb2g/33nxl+sU
sT6xNUGedZYJMBti3IZB7F2TqwX5azc9+2yirMojDJMwjsPZT2hTjjy40CmC5A1grEmaUv/gBlGa7BiqPAdhxMP
DIc+T7DdEQZdnzPPnCGfZrM4eMvNrlWhdAiBpSjn/AYegAlN2FWHafNUig/Lu8eBaHMcT2O2rDLpaDupsLFGBP9
YI4OiS6YhSmIt1iBDFVkTSHNrcUmLzgyC11e5nVWqb3wqd7+J+hitUrEpqqKMULb7XawbTcYyDzlqFk8PjHqPzJ
/qpt02SdkF0ditZw/z2dRsUiuyVvyejO/2b1Ldr9ffb0EHDVE97sorEBQTCh2qWsRd8zwGDOOTL7H8LQ5Y1Mv7f
bhvQsVrcxJCzeGn+R6nKyV0m2orXQ0/ht1QdgVui31P3TD/L6f+NdvFL+67e0hWeleGHr6pkPX9R3iMkoJHTHfd
xhjnuksA/WqiXXJyQgTIyc+rT9H/3fV0v7GoKmzmR89XOh4ur4B8cjnkw==</pareq>
              <acs-url>https://3dsecure-test.wirecard.com/acssim/app/bank</acs-url>
        </three-d>
</payment>
```

6.5 ACS Redirect

In addition to the interface setup between the merchant and Wirecard, the successful 3-D Secure implementation requires some interaction between the merchant and the issuing bank via the cardholder browser. **There is NO interaction with Wirecard for this step.** For the 3-D authentication to work, it is imperative that the merchant communicates with the ACS by SSL-encrypted HTTP POST request. In setting up the HTTPS Post authentication request, the merchant must ensure that following the enrollment check the purchase order is redirected from the merchant server to the card issuer's Access Control Server (ACS).

6.5.1 Fields

The following fields are specific to 3-D Secure.

Term	Man / Opt	Format	Max Size
PaReq	M	alphanumeric	16000
TermURL	M	alphanumeric	100
MD	M	alphanumeric	256

6.5.2 ACS HTTPS Redirect

In case a card is eligible and enrolled (‘check-enrollment’, Transaction Status Code = 200.0000) the merchant system must redirect the customer’s browser to the ACS URL provided in the Verify 3-D Participation response. If the card is not enrolled ((‘check-enrollment’, Transaction Status Code = 500.1072) , the ACS redirect must be skipped.

This HTTPS POST message includes the web address (URL) of the ACS and three hidden input types: <PaReq>, <TermUrl> and <MD>. The TermUrl defines the web address of the merchant site to which the issuer returns the Payment Authentication Response (PARes) message. The parameter type <MD> is reserved for specific merchant data. Although this field is mandatory, it does not need to have a value defined. If this input type is omitted an authentication error will occur and the payment process is aborted. The MD may be useful for retrieving transaction data from the database or recalling a transaction. The data is returned untouched by the ACS with the Payment Authentication Response (PARes).

Example: Auto submission POST Request

```
<html>
   <head>
      <meta HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8" />
      <meta HTTP-EQUIV="Cache-Control" CONTENT="no cache" />
      <meta HTTP-EQUIV="Pragma" CONTENT="no cache" />
      <meta HTTP-EQUIV="Expires" CONTENT="0" />
   </head>
```

```
    <body OnLoad="AutoSubmitForm();">
       <form name="downloadForm" action="AcsUrl" method="POST">
          <input type="hidden" name="PaReq" value="PaReq" />
          <input type="hidden" name="TermUrl" value="TermUrl" />
          <input type="hidden" name="MD" value="optionalValue" />
          <SCRIPT LANGUAGE="Javascript">
             <!--function AutoSubmitForm() { document.downloadForm.submit();}//-->
          </SCRIPT>
          <input type="submit" name="continue" value="Continue" />
       </form>
    </body>
 </html>
```

6.5.3 ACS HTTPS Redirect to TERM-URL

The cardholder's browser passes the encrypted PaReq message unprocessed to the ACS. This step is made up of two phases: The server invokes an authentication popup or inline window in the cardholder's browser. The cardholder enters a password in the authentication window and returns the data to the ACS.

The ACS authenticates the cardholder's password, constructs the verification ID, and posts an SSL-encrypted and digitally signed Payment Authentication Response (PARes) to the TermURL via the Account Holder's browser. Encryption and signature ensure that the content cannot be modified during transit.

The cardholder browser redirects the fully encrypted PARes to the merchant's server address specified as TermUrl. The response message contains the results of the cardholder authentication and the untouched merchant data (MD).

6.6 Check Payer Response

The merchant has received the PARes via the TermUrl. The PARes is a digitally signed XML document and has to be forwarded to Wirecard for validation. This is done with the Check Payer Response. This request should be used in case you are using the Payment Processing API as “MPI only”. Otherwise the PARes should be included along with the subsequent payment request.

NOTE: To ensure a unique transaction flow the Check Payer Response should be called for ALL 3-D transactions regardless of the enrollment status. The Check Payer Response includes the ECI which is needed for further payment processing.

6.6.1 Fields

See Section 0 for a description of the other payment fields. The following fields are specific to 3-D Secure.

Term	Man / Opt	Format	Max Size
three-d.pares	M	alphanumeric	16000

6.6.2 Sample XML Request and Response

Sample Request:

```
<?xml version="1.0" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
       <merchant-account-id>3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-account-id>
       <request-id>e0ea66d3-6368-44be-94af-ba91dfa84566</request-id>
       <transaction-type>check-payer-response</transaction-type>
       <parent-transaction-id>9e2c1d8a-a672-11e1-b76c-005056ab0016</parent-transaction-
id>
```

```
        <three-d>

        <pares>eJydV1mTqroWfvdXWH0ePXsziegu21NhEMEGRWbeEBBQBmWWX3+jdvfp6eHem6quTla+rCnfyp
L5P12aDJugKOM8e37CfqNPwyDzcj/OwucnXVv+mj79s5hrUREErBp4dREsBkM45lJQlm4YDGP/+YmczTCCoGbYe
DKZYegYw2bo0wN3x27BLijvSBQdkxiBzaZjbEaOJ/h09gl5R796s4DO/MbnyNvyM0gKCi9ys+qz+L7lehdakBcY
TozJyRx5XX7HpUEhsAsSRVEM/qFz5CH4DszcNFiYcRF4buEP2SDNh2qUn+fIfeM73svrrCquiykOzb8tvsPqIll
EVXX+gyBt2/5uXw389vJ0jtw2P0eM/BzyfFvfxOVPjnSxvzAIWXJ6+eriMmkkES6Zu6We5bh8BM9z5Ib4fs53q2
CBoxiOkjg5xCZ/UPIPis2Ru/yHhKe3GG/3BbP9mH8HnV/dBK/gW8K/yH7IZF0UkI4wlWMIf199BwbdOc8CqAMS5
n3+JX8/J2p+drNF92FAzwjomvuVcZr1g90qTn/M1F3+HV5WblWXC3uOvM5+CNltmgUAgGaRzwM8Bry0O+SHHHjx
AiVh/PD/z3pBEuZFXEXpLU2fBV9y9TFYmLlb/b7WvRqHGXS9CIbw5cjK56ePFCZ+50WI4LCaEHSGQIBfxuFfT4v
7qcAXskO+mDNulmex5yZx71awsqWginJ/+O7LTyq13U0rhuw45hdU+8vDxtmvmwQlMPIvEx5j8jSFl14+IYt/ff
xvVH/1tijdX2XkYjdFu+AQ3CgXDPWd8Pz018/P11wr3Kw85EVafpj/b1aDrAmS/Bz4v8o3528OIB9Vs3EYlNX/E
9NbPA8NhpvUwYJi4/O21LsC10t8vKHaptpnp+X6pHmQYx+Rc+Q9D3D+8SrfE/0AXo1Gvcy62TYPFXu/DlyfvOZH
yqt228rVkvqY4gBJsEuyr5gSWCdzE60Jed0Y/IwlkheZtxi9zA/4NkrUwcg7bAo72o9scflyTUuBTu2XU487+zN
Db9GDZiX9aYaxbYG0skmORSphEHKyF9HR6Jiglng1G4IxiHGCnAfrfJKMSbRy8pBmizVjKt1kFm30paTX1IpkrV
1B7GYID7DMpPjrGYRUhlKrU1MVoCuVUQacGqCUx9uiMbBND7QNNm4s68QiplKWMlbahwhvw4NQbIAillqX7L24u
zg+X+G6r3SWJVv4TmwI7lqc15RxXroradTuB6wfLxUXB+M0Q+nLniqyiDlABSSiLZXgINrrzQsFnp8fif+Q7Pk6
uD5uwSLRGetW7mOm1vtj4FXyrTcx8jPhl59ay9/Djf783swAD9fP5JiYoMQfFb5JwZB2s9MwPwwlt4b8iuty+FL
5fw+ZZ0mfI18N3C0yQVHFB1jRsDtIgsB5PcMArA5BK9AgFBhhbbesYovr3BGixpOBwi1pBbSh03ZMD0Q6lA0a2B
pIDE1SuXbV2qyhKOsB10WybcmoY4msbYkngUtqnzeu+3SJuuasts22k46gfSgotbWORR6xS7yTHHmpEqomeTtb2
9buPNjjZLRnaM0xZcxLDXaP7xKB85N9DDpJs1sZ3K2y3BX7bHQpJzZhXB1dbhxeDxVT7ga+mVwdSwh1woj3hHhy
TTLyb5sodD/yZIm1oWfcWNKEXtK43rzJNKmVegHKQC9rOmoe6UJSynbAKHfLPNeKsqFxhcScIBF1jomkWNaUHrb
qfpPIKAwDeixG+7QM97gdauYSRpKgrmX0AtehDsvpA4kWHoc76cXAoZfmIxeK7nW8BqxHrqBPrHzyriThWmJi48
urw5DHPY62YcjFEkB5Rr3wA1XYE6zC0UDRARgLNNuCG2ANcnizCqslHTndHSfICx53R1vNukNv9kQjVTi5O9FEX
1YncW+uMHd7cAaFz2+TaMLn+io6Bbv22GahjDJSma3VmOG1UTAGtKgJLyHRxAlFg6kk6ecpaYU4v3TaFilHlt6P
1+aVnA7w6fVkCkRvbJelispIWIVI7pL7ZEQSx+NlQlMro8TsZR9tNW7ZN4wqiEY2vcS9jng6O9qE/uxUAWQE62s
Q2b6UNEr7Ivp+bDWyVWuiMhL7zbLlN+qmYY8H7Vh2Z7ew9WQnLcena66fs9ySQpWq5Ehcp74mLJmcNqpBvNIZnd
nsE2PUsIqyrCXgrIUDzZ94PwScwqfbw9agJsfpcirW48CjOANltGhmqRSppqJ9aTQQSjQA/HEQBuMWPsTtnSI7V
KOB0AIWbG63uFKmNDhM4e1IDD0GLdfeMQkN2pYJbWHd2jSt6CtYhm24D/kBpIUvtCpdsgxduDx9nqj0Gc5zjXuR
wINzdCQxhiF1ggYOD6rkGs99L6vBW13dywoWjcSCN96dITH7W50qd17R4j6Dh9NbhSQNrN3ya0kNdrhRCyzXbXp
AvhFUOn22quPG0YdkVfDZyVFpdY/PUIGTEy/bnZ00OT6cGYcDxTJQl59dXWtH3sj6xuY3MtOvZAYKNEYIJpfnzG
7VrukZUWX20SvplWb2O3zrbPUBmMBOoO+IcwSfl5zaHzgG2xY1omm9xDPU2ZKmhXm4qsr1GPWF3OLkcqOPSpoc8
ScPDdWuPqn+KexVsK4G6joW906M1J6xchqE3p7XXU7bK+/iUFsaFfAXT+OaAwoaVvNqbRR1h8v1onhO6zNIsRIv
ByTLei/IyeI0kOXY0qly3FxwQGIXbNZ0Unrow2lMwc7nTB3noE0tKX3plxzPtEad7PwtZlHAwXCEzflDL1MpIzT
woVYHRirlkwa1ZgYTkVawPdTAVbd5N1udcXojo3qnURuZ2oV1YDa1MdqWE43rpqsindRlNxKZaNRcidTC+7AbZG
QiPFrFx87wkDz6FPLeu/7taovbF879m3IxgL97Pn5x/geuCnnt</pares>
        </three-d>
</payment>
```

Sample Response:

```
<?xml version="1.0" standalone="yes"?>
<payment    self="http://10.0.0.142:8080/engine/rest/merchants/3e549fd0-50de-11e0-
b8af-0800200c9a66/payments/9ef2fd42-a672-11e1-b76c-005056ab0016"
xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id
ref="http://10.0.0.142:8080/engine/rest/merchants/3e549fd0-50de-11e0-b8af-
0800200c9a66">3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-account-id>
        <transaction-id>9ef2fd42-a672-11e1-b76c-005056ab0016</transaction-id>
        <request-id>e0ea66d3-6368-44be-94af-ba91dfa84566</request-id>
        <transaction-type>check-payer-response</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2012-05-25T14:05:02.748Z</completion-time-stamp>
        <statuses>
                <status  code="201.0000"  description="3d-acquirer:The  resource  was
successfully created." severity="information"/>
                <status     code="200.1083"     description="3d-acquirer:Cardholder
Successfully authenticated." severity="information"/>
        </statuses>
```

```
      <requested-amount currency="USD">1.000000</requested-amount>
      <parent-transaction-id>9e2c1d8a-a672-11e1-b76c-005056ab0016</parent-
transaction-id>
      <authorization-code/>
      <three-d>
            <eci>05</eci>
            <xid>V3NMZzNya2N5Vlh2MWRFUno2NjA=</xid>
            <cardholder-authentication-
value>AAABD///////////////AAAAAAA=</cardholder-authentication-value>
            <cardholder-authentication-status>Y</cardholder-authentication-
status>
      </three-d>
</payment>
```

6.7 Payment Request with PARes

The merchant has received the PARes via the TermUrl. The PARes is forwarded to Wirecard along with payment request.

Term	Man / Opt	Format	Max Size
parent-transaction-id	M	alphanumeric	36
three-d.pares	M	alphanumeric	16000

The parent-transaction-id supplied is the transaction-id of the ‘check-enrollment’ response. The pares is received after the dialogue with the issuing bank.

Sample Request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
      <merchant-account-id>3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-
account-id>
      <request-id>fbeee5af-8329-48ad-af9e-3aaca489520a</request-id>
      <transaction-type>purchase</transaction-type>
      <parent-transaction-id>dcc9947c-74e2-11e0-877b-005056ab0016</parent-
transaction-id>
      <three-d>
            <pares>eJxtk11PgzAUhu</pares>
      </three-d>
</payment>
```

Sample Response:

```
<?xml version="1.0"?>

<payment xmlns:ns2="http://www.elastic-payments.com/schema/payment"
xmlns:ns3="http://www.elastic-payments.com/schema/config" self="
https://localhost/engine/rest/merchants/3e549fd0-50de-11e0-b8af-
0800200c9a66/payments/dd363df2-74e2-11e0-877b-005056ab0016">

      <merchant-account-id
ref="https://10.0.0.143/engine/rest/merchants/3e549fd0-50de-11e0-b8af-
0800200c9a66">3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-account-id>

      <transaction-id>dd363df2-74e2-11e0-877b-005056ab0016</transaction-id>

      <request-id>fbeee5af-8329-48ad-af9e-3aaca489520a</request-id>

      <transaction-type>purchase</transaction-type>

      <transaction-state>success</transaction-state>
      <completion-time-stamp>2011-05-02T16:47:08.264Z</completion-time-stamp>

      <statuses>

             <status severity="information" description="wirecard:The request
completed successfully." code="200.0000"/>

             <status severity="information" description="wirecard: 3D: Cardholder
Successfully authenticated." code="200.1083"/>

      </statuses>

      <requested-amount currency="EUR">1.000000</requested-amount>

      <parent-transaction-id>dcc9947c-74e2-11e0-877b-005056ab0016</parent-
transaction-id>

      <auth-code>077863</auth-code>

      <three-d>

             <cardholder-authentication-status>Y</cardholder-authentication-
status>

      </three-d>

</payment>
```

6.8 Payment Request with 3rd Party MPI

This scenario is applicable when the merchant uses an external MPI in conjunction with Wirecard as a gateway to the acquiring network. The protocol is exactly the same as defined for authorization or purchase request above. Only the relevant 3-D artifacts should be included in addition, conditional on their availability. The response remains same as defined above.

Term	Man / Opt	Format	Max Size
three-d.eci	Conditional	numeric	2-digits
three-d.xid	Conditional	alphanumeric	28
three-d.cardholder-authentication-value	Conditional	alphanumeric	16000

Sample Request:

```
<?xml version="1.0"?>

<payment xmlns="http://www.elastic-payments.com/schema/payment">

      <merchant-account-id>3e549fd0-50de-11e0-b8af-0800200c9a66</merchant-
account-id>

      <request-id>fbeee5af-8329-48ad-af9e-3aaca489520a</request-id>

      <transaction-type>purchase</transaction-type>

      <parent-transaction-id>dcc9947c-74e2-11e0-877b-005056ab0016</parent-
transaction-id>

      <three-d>

             <eci>05</eci>

             <xid>N1dYQmpuYlVUZ2hZNUhRZHdnQW0</xid>

             <cardholder-authentication-
```

```
value>Ty56UNQmpuYlVUZ2hZNUhRZ09UYVF7</cardholder-authentication-value>
          </three-d>
</payment>
```

7 Online Banking Payments

Merchants can send payments to the Wirecard Payments Platform using various payment methods. In order to be able to create a payment using a specific payment method, the merchant has to have a valid subscription for this payment method and needs to specify it correctly inside the payment request.

7.1 Create a Payment for Online Banking Methods

URL	https://hostname/engine/rest/paymentmethods
Request Formats	XML
Response Formats	XML
Request Methods	POST
Transaction Type	DEBIT

For using online banking payment methods (for which the end customer will be forwarded to a landing page on which he will be able to execute his payment) the requests have to be sent to : https://hostname/engine/rest/paymentmethods/.

The transaction type within the request must be *DEBIT.*
If the request is successful the *Forward-URL* to the Landing-Page will be sent in the response.
The notification allows a merchant to receive the final status of a payment as soon as Wirecard receives it from the 3rd party. Usually, this status is received within seconds of the completion of the transaction, however, may take up to a few days depending on the respective payment method.
Once Wirecard has received a notification from the 3rd party about the final status of the transaction this status will be communicated to the merchant via the NotificationURL that is configured in the merchant account.The merchant can also send a NotificationURL in each request dynamically. This will overwrite the NotificationURL that is configured in the merchant account. If no NotificationURL is configured in the merchant account or if no NotificationURL is sent in the request the notification will be sent via Email to the merchant in case the Email address has been configured.

7.2 Direktüberweisung

Please see also Chapter 6 Appendix A: Payment Methods.

The payment method “Direktüberweisung” is an Online Banking Payment Method and works like described above.

The notification for the final status of a payment should be received after 25 Minutes for a “Direktüberweisung” transaction. If the notification is not sent within this time, the merchant has the possibility to send a request for “Retrieve Transaction by Transaction ID” (see Chapter 0) or “Retrieve Transaction by Request ID” (see Chapter 5.4). With the response the merchant will get the current status of the requested transaction.

5.2.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method Direktüberweisung:

Term	Man /Opt	Format	Max Size	Description
account-holder.address.city	O	Alphanumeric	32	This is the end-consumer’s city
account-holder.address.country	O	Alphanumeric	3	This is the end-consumer’s country
account-holder.address.postal-code	O	Alphanumeric	16	This is the end-consumer’s postal code
account-holder.address.state	O	Alphanumeric	32	This is the end-consumer’s state
account-holder.address.street1	O	Alphanumeric	128	This is the first part of the end-consumer’s street
account-holder.address.street2	O	Alphanumeric	128	This is the second part of the end-consumer’s street.
account-holder.date-of-birth	O	Date	0	This is the end-consumer’s birthdate
account-holder.email	O	Alphanumeric	64	This is the end-consumer’s Email-Address
account-holder.first-name	M	Alphanumeric	32	This is the first name of the end-consumer
account-holder.gender	O	Alphanumeric	1	This is the end-consumer’s gender
account-holder.last-name	M	Alphanumeric	32	This is the last name of the end-consumer
account-holder.phone	O	Alphanumeric	32	This is the phone number of the end-consumer
bank-account.account-number	M	Alphanumeric	34	This is the account number of the end.consumer.
bank-account.bank-code	M	Alphanumeric	15	This is the bank code of the bank of the end-consumer.
custom-field.field-name	O	Alphanumeric	36	This is the name for the custom field
custom-field.field-value	O	Alphanumeric	256	This is the content of the custom field. In this field the merchant can send additional information

descriptor	O	Alphanumeric	27	Description on the settlement of the account holder’s account about a transaction
ip-address	O	Alphanumeric	15	The global (internet) IP address of the customer computer.
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
notification.transaction-state	O	Alphanumeric	12	This is the status of a transaction send in the notification
order-detail	O	Alphanumeric	65535	This is a field for details of an order filled by the merchant
order-number	O	Alphanumeric	64	This is the order number of the merchant
parent-transaction-id	O	Alphanumeric	36	Transaction ID of the first transaction of a payment
payment-methods.payment-method-name	M	Alphanumeric	15	This is the name of the payment method that that is chosen from the end-consumer
processing-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be redirected after he has fulfilled his payment.This is normally a page on the merchant’s website.
notifications/notification/@url	O	Alphanumeric	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
requested-amount	M	Numeric	18,2	This is the amount of the transaction
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
status.code	M	Alphanumeric	12	This is the code of the status of a transaction

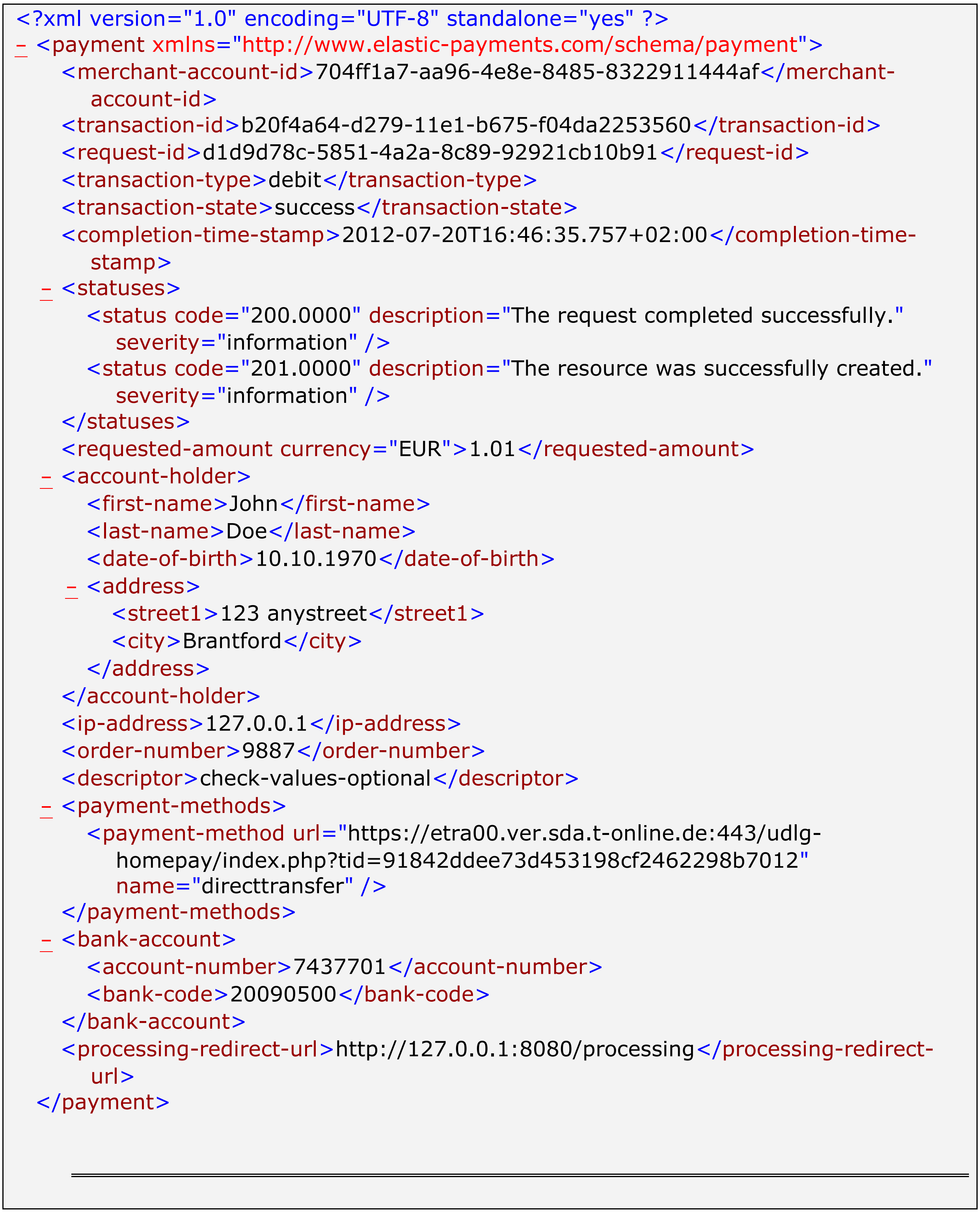
status.description	M	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M	Alphanumeric	12	This is the status of a transaction
transaction-id	M	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It ist generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For DÜ only DEBIT is allowed.

5.2.2 XML- Samples for Direktüberweisung

Sample of a successful Direktüberweisung request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>704ff1a7-aa96-4e8e-8485-8322911444af</merchant-
      account-id>
    <request-id>d1d9d78c-5851-4a2a-8c89-92921cb10b91</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="EUR">1.01</requested-amount>
  - <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <date-of-birth>10.10.1970</date-of-birth>
    - <address>
        <street1>123 anystreet</street1>
        <city>Brantford</city>
      </address>
    </account-holder>
  - <bank-account>
      <account-number>7437701</account-number>
      <bank-code>20090500</bank-code>
    </bank-account>
  - <payment-methods>
      <payment-method name="directtransfer" />
    </payment-methods>
    <order-number>9887</order-number>
    <descriptor>check-values-optional</descriptor>
     <processing-redirect-url>http://127.0.0.1:8080/processing</processing-redirect-
      url>
    <ip-address>127.0.0.1</ip-address>
  </payment>
```

Sample of a successful Direktüberweisung response:



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>704ff1a7-aa96-4e8e-8485-8322911444af</merchant-
      account-id>
    <transaction-id>b20f4a64-d279-11e1-b675-f04da2253560</transaction-id>
    <request-id>d1d9d78c-5851-4a2a-8c89-92921cb10b91</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2012-07-20T16:46:35.757+02:00</completion-time-
      stamp>
  - <statuses>
      <status code="200.0000" description="The request completed successfully."
        severity="information" />
      <status code="201.0000" description="The resource was successfully created."
        severity="information" />
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
  - <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <date-of-birth>10.10.1970</date-of-birth>
    - <address>
        <street1>123 anystreet</street1>
        <city>Brantford</city>
      </address>
    </account-holder>
    <ip-address>127.0.0.1</ip-address>
    <order-number>9887</order-number>
    <descriptor>check-values-optional</descriptor>
  - <payment-methods>
      <payment-method url="https://etra00.ver.sda.t-online.de:443/udlg-
        homepay/index.php?tid=91842ddee73d453198cf2462298b7012"
        name="directtransfer" />
    </payment-methods>
  - <bank-account>
      <account-number>7437701</account-number>
      <bank-code>20090500</bank-code>
    </bank-account>
    <processing-redirect-url>http://127.0.0.1:8080/processing</processing-redirect-
      url>
  </payment>
```

Sample of a failed Direktüberweisung request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```

```
- <payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>704ff1a7-aa96-4e8e-8485-8322911444af</merchant-
       account-id>
    <request-id>86fbf5c8-2b7d-4c0d-86a0-eed1b990f77f</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="EUR">1.01</requested-amount>
  - <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <date-of-birth>10.10.1970</date-of-birth>
    - <address>
        <street1>123 anystreet</street1>
        <city>Brantford</city>
      </address>
    </account-holder>
  - <bank-account>
      <account-number>4444333322221111</account-number>
      <bank-code>20090500</bank-code>
    </bank-account>
  - <payment-methods>
      <payment-method name="directtransfer" />
    </payment-methods>
    <order-number>16107</order-number>
    <descriptor>error_0</descriptor>
      <processing-redirect-url>http://127.0.0.1:8080</processing-redirect-url>
    <ip-address>127.0.0.1</ip-address>
  </payment>
```

Sample of a failed Direktüberweisung response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>704ff1a7-aa96-4e8e-8485-8322911444af</merchant-
       account-id>
    <transaction-id>b56c5f89-d279-11e1-b675-f04da2253560</transaction-id>
    <request-id>86fbf5c8-2b7d-4c0d-86a0-eed1b990f77f</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>failed</transaction-state>
    <completion-time-stamp>2012-07-20T16:46:41.395+02:00</completion-time-
       stamp>
  - <statuses>
      <status code="500.1050" description="Provider had a system error. Please try
         again later." severity="error" />
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
  - <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <date-of-birth>10.10.1970</date-of-birth>
    - <address>
        <street1>123 anystreet</street1>
```

```
          <city>Brantford</city>
        </address>
      </account-holder>
      <ip-address>127.0.0.1</ip-address>
      <order-number>16107</order-number>
      <descriptor>error_0</descriptor>
    – <payment-methods>
        <payment-method name="directtransfer" />
      </payment-methods>
    – <bank-account>
        <account-number>4444333322221111</account-number>
        <bank-code>20090500</bank-code>
      </bank-account>
      <processing-redirect-url>http://127.0.0.1:8080</processing-redirect-url>
        </payment>
```

Sample of a notification for a failed Direktüberweisung transaction:

```
<?xml version="1.0" encoding="UTF-8" ?>
– <payment xmlns="http://www.elastic-payments.com/schema/payment">
      <merchant-account-id>704ff1a7-aa96-4e8e-8485-8322911444af</merchant-account-id>
      <transaction-id>b602e44d-d279-11e1-b675-f04da2253560</transaction-id>
      <request-id>288eaa56-934e-4ee7-995e-bf60f53a9189</request-id>
      <transaction-type>debit</transaction-type>
      <transaction-state>failed</transaction-state>
      <completion-time-stamp>2012-07-20T16:46:42.302+02:00</completion-time-stamp>
    – <statuses>
        <status code="500.1999" description="directtransfer:The acquirer returned an unknown
          response. Contact Technical Support." severity="warning" />
      </statuses>
      <requested-amount currency="EUR">1.01</requested-amount>
      <ip-address>127.0.0.1</ip-address>
      <order-number>23233</order-number>
      <descriptor>status_wrongresp</descriptor>
    – <payment-methods>
        <payment-method name="directtransfer" />
      </payment-methods>
    – <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      – <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
            20010315" />
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
        – <Reference URI="">
          – <Transforms>
              <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
                signature" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <DigestValue>rqDlW1dmCaOcmzGatkW7sjc5kA0=</DigestValue>
          </Reference>
        </SignedInfo>

          <SignatureValue>epUaP8or3Y3ozKb2DgvPHikqhh2E8QTToh8IF2yOe0254oWA+rEEID
          01TyMOCWw0a5bQNlBUNx3V
```

```
            0inZ9DOm5U9xfT6vEPiU1lN+ekUCtnFXnFj/W5F6k8qZ9EqFZ9yk1DKdM+7IwuFfAyg52
            ObHEiRM
            b1sICqaX/RwqU+K3sNeVE3yw4YUyIrso0DNJLLap/w6i4/bjMOdptHqOD+5pTCpDKb7m
            VnLHwpF6
            FFszWCXssCZM99ILWBG6ycXSCZ6Pok8wSuDykbv8fRmXk+PTd/efnDzUFLIdq7x/Op1R
            6qj1plU3 Edbo43BdiuZyRj3KAjkAJco9VGwJpz7UPzoYPA==</SignatureValue>
    - <KeyInfo>
      - <X509Data>
          <X509SubjectName>CN=xxx,OU=Operations,O=Wirecard Elastic
             Payments,L=xxx,ST=ON,C=CA</X509SubjectName>

             <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6M
             QswCQYDVQQGEwJDQTELMAkGA1UE
             CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN
             0aWMgUGF5bWVu
             dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHh
             cNMTEwNjI0MDQ0
             NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECB
             MCT04xEDAOBgNVBAcT
             B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEz
             ARBgNVBAsTCk9w
             ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQ
             UAA4IBDwAwggEK
             AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4
             HeonfTxCN
             etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5
             d8RT5HaJV
             4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIl
             AMGzW7VgKD
             2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnR
             WbIzyQuyUyDeUb
             QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvc
             NAQEFBQADggEB
             ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbO
             D3HBXoyovZr
             mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SS
             ucBfIS5TJ54e7m
             02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8K
             CHpfw2j8G+dDgU
             jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13
             AzYwDP3 x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
        </X509Data>
      </KeyInfo>
    </Signature>
  </payment>
```

7.3 Sofortüberweisung

Please see also Chapter 6 Appendix A: Payment Methods.

The payment method "sofortüberweisung" is an Online Banking Payment Method and works like described above.

The notification for the final status of a payment should be received after 25 Minutes for a “Sofortüberweisung” transaction. If the notification is not sent within this time, the merchant has the possibility to send a request for “Retrieve Transaction by Transaction ID” (see Chapter 0) or “Retrieve Transaction by Request ID” (see Chapter 5.4). With the response the merchant will get the current status of the requested transaction.

5.3.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method Sofortüberweisung:

Term	Man /Opt	Format	Max Size	Description
account-holder.address.city	O	Alphanumeric	32	This is the end-consumer’s city
account-holder.address.country	O	Alphanumeric	3	This is the end-consumer’s country
account-holder.address.postal-code	O	Alphanumeric	16	This is the end-consumer’s postal code
account-holder.address.state	O	Alphanumeric	32	This is the end-consumer’s state
account-holder.address.street1	O	Alphanumeric	128	This is the first part of the end-consumer’s street
account-holder.address.street2	O	Alphanumeric	128	This is the second part of the end-consumer’s street.
account-holder.date-of-birth	O	Date	0	This is the end-consumer’s birthdate
account-holder.email	O	Alphanumeric	64	This is the end-consumer’s Email-Address
account-holder.first-name	M	Alphanumeric	27	This is the first name of the end-consumer. The maximum size of firstname and last-name in combination is 27 characters.
account-holder.gender	O	Alphanumeric	1	This is the end-consumer’s gender
account-holder.last-name	M	Alphanumeric	27	This is the last name of the end-consumer. The maximum size of firstname and last-name in combination is 27 characters
account-holder.phone	O	Alphanumeric	32	This is the phone number of the end-consumer

bank-account.account-number	O	Alphanumeric	34	This is the account number of the end consumer. This parameter has to be filled in case this element is sent in the request. In case it is a foreign bank account the IBAN has to be filled in.
bank-account.bank-code	O	Alphanumeric	15	This is the bank code of the bank of the end-consumer. This parameter has to be filled in case this element is sent in the request. In case it is a foreign bank account the BIC has to be filled in.
bank-account.bic	O	Alphanumeric	15	This is the Business Identifier Code of the bank of the end-consumer. This parameter has to be filled in case this element is sent in the request. Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})?)
bank-account.iban	O	Alphanumeric	34	This is the International Bank Account Number of the end-consumer. This parameter has to be filled in case this element is sent in the request. Allowed characters: [a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}
custom-field.field-name	O	Alphanumeric	36	This is the name for the custom field

custom-field.field-value	O	Alphanumeric	256	This is the content of the custom field. In this field the merchant can send additional information
descriptor	M	Alphanumeric	27	Description on the settlement of the account holder's account about a transaction. The following characters are allowed: umlaut, - '0-9','a-z','A-Z',' ' , '+',',','-','.'
ip-address	O	Alphanumeric	15	The global (internet) IP address of the customer computer.
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
notification.transaction-state	O	Alphanumeric	12	This is the status of a transaction send in the notification
order-detail	O	Alphanumeric	65535	This is a field for details of an order filled by the merchant
order-number	M	Alphanumeric	64	This is the order number of the merchant
parent-transaction-id	O	Alphanumeric	36	Transaction ID of the first transaction of a payment
payment-methods.payment-method-name	O	Alphanumeric	15	This is the name of the payment method that that is chosen from the end-consumer
processing-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be redirected after he has fulfilled his payment.This is normally a page on the merchant's website.

success-redirect-url	M	Alphanumeric	256	The URL to which the consumer will be re-directed after a successful payment. This is normally a success confirmation page on the merchant’s website.
cancel-redirect-url	M	Alphanumeric	256	The URL to which the consumer will be re-directed after he has cancelled a payment. This is normally a page on the merchant’s website
notifications/notification/@url	O	Alphanumeric	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
requested-amount	M	Numeric	18,2	This is the amount of the transaction. The amount of a sofortüberweisung transaction must be >=0,20 €.
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
status.code	M	Alphanumeric	12	This is the code of the status of a transaction
status.description	M	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M	Alphanumeric	12	This is the status of a transaction

transaction-id	M	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It ist generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For DÜ only DEBIT is allowed.

5.3.2 XML- Samples for Sofortüberweisung

Sample of a successful sofortüberweisung request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
  account-id>
  <request-id>2dd77e58-57c7-4363-9679-7073814cb81a</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="EUR">2.32</requested-amount>
  <payment-methods><payment-method name="sofortbanking"/></payment-
  methods>
  <descriptor>test</descriptor>
  <success-redirect-url>http://127.0.0.1:8080/success</success-redirect-url>
  <cancel-redirect-url>http://127.0.0.1:8080/cancel</cancel-redirect-url>
  <account-holder>
            <first-name>John</first-name>
            <last-name>Doe</last-name>
  </account-holder>
  <bank-account>
            <account-number>4444333322221111</account-number>
            <bank-code>20090500</bank-code>
            <bic>DEUTDEFF</bic>
            <iban>1234567899940</iban>
  </bank-account>
</payment>
```

Sample of a successful sofortüberweisung response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
          <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
account-id>
   <transaction-id>01c9e586-f1d6-11e1-b0a2-f04da2253560</transaction-id>
   <request-id>2dd77e58-57c7-4363-9679-7073814cb81a</request-id>
   <transaction-type>debit</transaction-type>
   <transaction-state>success</transaction-state>
         <completion-time-stamp>2012-08-29T14:35:23.108+02:00</completion-time-
stamp>
   <statuses>
```

```
     <status code="201.0000" description="The resource was successfully created."
severity="information"/>
   </statuses>
   <requested-amount currency="EUR">2.32</requested-amount>
   <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
   </account-holder>
   <descriptor>test</descriptor>
   <payment-methods>
        <payment-method  url="https://www.sofort.com/payment/go/4551-5272-5134-
7456" name="sofortbanking"/>
   </payment-methods>
   <bank-account>
      <account-number>4444333322221111</account-number>
      <bank-code>20090500</bank-code>
      <iban>1234567899940</iban>
      <bic>DEUTDEFF</bic>
   </bank-account>
   <cancel-redirect-url>http://127.0.0.1:8080/cancel</cancel-redirect-url>
   <success-redirect-url>http://127.0.0.1:8080/success</success-redirect-url>
</payment>
```

Sample of a failed sofortüberweisung request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
account-id>
  <request-id>b438ae34-c31c-44c2-831d-f58a8a0748c0</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="EUR">2.32</requested-amount>
   <payment-methods><payment-method  name="sofortbanking"/></payment-
methods>
  <descriptor>test</descriptor>
  <success-redirect-url>http://127.0.0.1:8080/success</success-redirect-url>
  <cancel-redirect-url>http://127.0.0.1:8080/cancel</cancel-redirect-url>
  <fail-redirect-url>http://127.0.0.1:8080/fail</fail-redirect-url>
  <account-holder>
           <first-name>John</first-name>
           <last-name>Doe</last-name>
  </account-holder>
  <bank-account>
           <account-number>4444333322221111</account-number>
           <bank-code>20090500</bank-code>
           <bic>DEUTDEFF</bic>
           <iban>1234567899940</iban>
  </bank-account>
</payment>
```

Sample of a failed sofortüberweisung response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
account-id>
   <request-id>b438ae34-c31c-44c2-831d-f58a8a0748c0</request-id>
   <transaction-type>debit</transaction-type>
   <requested-amount currency="EUR">2.32</requested-amount>
    <payment-methods><payment-method  name="sofortbanking"/></payment-
methods>
   <descriptor>test</descriptor>
   <success-redirect-url>http://127.0.0.1:8080/success</success-redirect-url>
   <cancel-redirect-url>http://127.0.0.1:8080/cancel</cancel-redirect-url>
   <fail-redirect-url>http://127.0.0.1:8080/fail</fail-redirect-url>
   <account-holder>
               <first-name>John</first-name>
               <last-name>Doe</last-name>
   </account-holder>
   <bank-account>
               <account-number>4444333322221111</account-number>
               <bank-code>20090500</bank-code>
               <bic>DEUTDEFF</bic>
               <iban>1234567899940</iban>
   </bank-account>
</payment>
```

Sample of a notification for a successful sofortüberweisung transaction:

```
<?xml version="1.0" encoding="UTF-8"?><payment
 xmlns="http://www.elastic-payments.com/schema/payment">
            <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
account-id>
    <transaction-id>73590625-f1d6-11e1-b0a2-f04da2253560</transaction-id>
    <request-id>ee409c23-5dad-4bc2-ad4e-fc34065093c0</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>success</transaction-state>
          <completion-time-stamp>2012-08-29T14:38:33.560+02:00</completion-time-
stamp>
    <statuses>
         <status  code="201.1126"  description="sofortbanking:Successful  confirmation
received from the bank." severity="information"/>
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
    <descriptor>status_received_credited</descriptor>
    <payment-methods>
        <payment-method name="sofortbanking"/>
    </payment-methods>
<Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMetho
d                                  Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/><SignatureMethod  Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
```

```
sha1"/><Reference                                URI=""><Transforms><Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>nCQV1V6hm
NKZF4aaej1i6Hsshgc=</DigestValue></Reference></SignedInfo><SignatureValue>b
Oe6398GUPmi/VwAWfcb/MSTXxp4jYBGhyYZbW1s9gqmKwiDvNNGOXC8S7mLraxKifixG
mX0S8KQ
gKH585Jcwne5+GWbLPDK3W/MZ3WaAGn7wr3jylbNae2H2gpZ5x3nAA/0bqjFafF85b5jn
M8V49uB
TIWZz04t9A2pTQiyP3Mzx4qan37rwyrm0hVUwGGziscVU6N5EfNf/xqEqH9Zr7Y55CnABQ6
dMuTR
f49s92XIXEcAolOtnUnSvdpHl1PEx83MmTgOZcK7hDtH1T3LdRm+DNh1Lk0IdKUjhWiGLV
pzs1ah
tVBgcHMyNReJFE+9XsvEJfyeQabhuDvxPEQLdw==</SignatureValue><KeyInfo><X509
Data><X509SubjectName>CN=Manoj      Sahu,OU=Operations,O=Wirecard      Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCA
ligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1U
E
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMg
UGF5bWVu
dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTE
wNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04
xEDAOBgNVBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgN
VBAsTCk9w
ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4I
BDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfT
xCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5
HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW
7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQ
uyUyDeUb
QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEF
BQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBX
oyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5
TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8KCHpfw2
j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwD
P3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate></X509Data></K
eyInfo></Signature></payment>
```

Sample of a notification for a failed sofortüberweisung transaction:

```
<?xml version="1.0" encoding="UTF-8"?><payment
xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
```

```
account-id>
    <transaction-id>0765088a-f1d6-11e1-b0a2-f04da2253560</transaction-id>
    <request-id>793f2bf5-79a6-43c0-add6-34e9393338fb</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>failed</transaction-state>
          <completion-time-stamp>2012-08-29T14:35:32.433+02:00</completion-time-
stamp>
    <statuses>
          <status code="500.1127" description="sofortbanking:Failed confirmation
received from the bank." severity="error"/>
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
    <descriptor>status_loss_notcredited</descriptor>
    <payment-methods>
        <payment-method name="sofortbanking"/>
    </payment-methods>
<Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMetho
d Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/><SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/><Reference URI=""><Transforms><Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>PZLE514wGO
wOhK5tfyAKeGfyg0c=</DigestValue></Reference></SignedInfo><SignatureValue>TY
2i/KdIXRW2xlhK/fRmY9SIXNB+xG+ElLmDecaUuUOrGEnYHZzCgdQTY2BKFtuMMnu7h4B
6eREw
vfmhwX79kP386DUrsxJjypkAbr/Q5w35eVdrOXGsKyBN3FZWXmfOAgXghZa0y4MXKPMq
YweUHakd
9+y+ToQsvjuKTuoxdgOyZ0d6dUMBI1v0GG3o1YaGE1WYRaZjZdUA6mvVzXzg23QG9QYt
0pt5koUz
Zvkx1k2hEdonWQ2CebwOfLWWK/fmXmZDiCuxhQiU4R05Y9yABpR6OqkfN9wCmV928at
NV8sTPyXQ
5ttJshdWqxQkNDHwB8YLBCEUY3OTBQBsLbZCjw==</SignatureValue><KeyInfo><X50
9Data><X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCA
ligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1U
E
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMg
UGF5bWVu
dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTE
wNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04
xEDAOBgNVBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgN
VBAsTCk9w
ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4I
BDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfT
xCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5
HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW
7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQ
uyUyDeUb
QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEF
BQADggEB
```

```
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBX
oyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5
TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8KCHpfw2
j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwD
P3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate></X509Data></K
eyInfo></Signature></payment>
```

8 Alternative Schemes Payments

Merchants can send payments to the Wirecard Payments Platform using various payment methods. In order to be able to create a payment using a specific payment method, the merchant has to have a valid subscription for this payment method and needs to specify it correctly inside the payment request. This chapter contains all payment methods that belong to the Alternative Schemes Payments like Wallet, Voucher and Online/Offline payments.

8.1 Create a Payment for Wallet Payment Methods

URL	https://hostname/engine/rest/paymentmethods
Request Formats	XML
Response Formats	XML
Request Methods	POST

For using wallet payment methods (for which the end customer will be forwarded to the landing page of the Wallet provider on which he will be able to execute his payment) the requests have to be sent to : https://hostname/engine/rest/paymentmethods/.

8.2 Skrill Wallet

Please see also Chapter 8 Appendix A: Payment Methods.

The transaction type within the request must be *DEBIT.*
If the request is successful the *Forward-URL* to the Landing-Page of the wallet provider will be sent in the response. The notification allows a merchant to receive the final status of a payment as soon as Wirecard receives it from the 3rd party. Usually, this status is received within seconds of the completion of the transaction, however, may take up to a few days depending on the respective payment method.

Once Wirecard has received a notification from the 3rd party about the final status of the transaction this status will be communicated to the merchant via the NotificationURL that is configured in the merchant account.If no notificationURL is configured in the merchant account the notification will be sent via Email to the merchant in case the Email address has been configured. In case the notification hasn't received within 40 minutes the merchant can request the status of the transaction by sending a “Retrieve Transaction by Transaction ID” (see Chapter 0) or “Retrieve Transaction by Request ID” (see Chapter 5.4).

If the merchant wants to offer other payment methods to the end-consumer instead inside the Skrill Wallet (e.g. in case the wallet has not enough balance to pay the goods or services) the account of the merchant at Skrill must be activated for the Flexible Split Gateway. That means in case the merchant has submitted a payment method it is preselected for the customer upon registration or login. All other payment methods enabled for the merchant for the customers' registration country are available. If the payment method is not supported by registration country of the customer, then a message will apear indicating the payment type is not supported, showing all other available payment methods for their country.

If the Flexible Split Gateway is activated and the end-customer can also choose other payment methods within Skrill debit returns can occur (e.g. if the end customer chooses credit card or direct debit as prefered payment method). This debit-return will be sent from Wirecard to the merchant within an additional notification.

6.2.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method Skrill Wallet:

Term	Man /Opt	Format	Max Size	Description
account-holder.address.city	O	Alphanumeric	32	This is the end-consumer's city
account-holder.address.country	O	Alphanumeric	3	This is the end-consumer's country
account-holder.address.postal-code	O	Alphanumeric	16	This is the end-consumer's postal code
account-holder.address.state	O	Alphanumeric	32	This is the end-consumer's state
account-holder.address.street1	O	Alphanumeric	128	This is the first part of the end-consumer's street
account-holder.address.street2	O	Alphanumeric	128	This is the second part of the end-consumer's street.
account-holder.date-of-birth	O	Date	0	This is the end-consumer's birthdate
account-holder.email	M/O	Alphanumeric	64	This is the end-consumer's Email-Address. It is mandatory if the field wallet-account-id is not sent in the initial request.
account-holder.first-name	M	Alphanumeric	27	This is the first name of the end-consumer. The maximum size of firstname and last-name in combination is 27 characters.
account-holder.gender	O	Alphanumeric	1	This is the end-consumer's gender

account-holder.last-name	M	Alphanumeric	27	This is the last name of the end-consumer. The maximum size of firstname and last-name in combination is 27 characters
account-holder.phone	O	Alphanumeric	32	This is the phone number of the end-consumer
wallet-account-id	M/O	Alphanumeric	50	The unique identifier of the Account Holder's Wallet Account. It is mandatory if the field account-holder.email is not sent in the intitial request.
Locale	M	Alphanumeric	6	Code of the language. Can be any of en,de,es,fr,it,pl,gr,ro, ru,tr,cn,cz,nl, da,sv or fi. Can be sent in the format <language> or in the format <language_country>.
custom-field.field-name	O	Alphanumeric	36	This is the name for the custom field
custom-field.field-value	O	Alphanumeric	256	This is the content of the custom field. In this field the merchant can send additional information
descriptor	M	Alphanumeric	27	Description on the settlement of the account holder’s account about a transaction. The following characters are allowed: umlaut, - '0-9','a-z','A-Z',' ' , '+',',','-','.'
ip-address	O	Alphanumeric	15	The global (internet) IP address of the customer computer.
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account

notification.transaction-state	O	Alphanumeric	12	This is the status of a transaction send in the notification
order-detail	O	Alphanumeric	65535	This is a field for details of an order filled by the merchant
order-number	M	Alphanumeric	64	This is the order number of the merchant
parent-transaction-id	O	Alphanumeric	36	Transaction ID of the first transaction of a payment
payment-methods.payment-method-name	O	Alphanumeric	15	This is the name of the payment method that that is chosen from the end-consumer
processing-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be redirected after he has fulfilled his payment.This is normally a page on the merchant’s website.
success-redirect-url	M	Alphanumeric	256	The URL to which the consumer will be re-directed after a successful payment. This is normally a success confirmation page on the merchant’s website.
cancel-redirect-url	M	Alphanumeric	256	The URL to which the consumer will be re-directed after he has cancelled a payment. This is normally a page on the merchant’s website
notifications/notification/@url	O	Alphanumeric	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.

requested-amount	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
settlement-gross-amount	M	Numeric	18,3	The amount due to the Merchant Account for this Payment, in the Payment Settlement Currency, gross of rates, fees, and reserve. The amount of the decimal place is dependent of the currency.
settlement-gross-currency	M	Alphanumeric	3	The currency in which the Merchant Account can expect his Payment Settlement Amount.
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
status.code	M	Alphanumeric	12	This is the code of the status of a transaction
status.description	M	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M	Alphanumeric	12	This is the status of a transaction
transaction-id	M	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It ist generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For Skrill Wallet only DEBIT is allowed in the initial request.

6.2.2 XML- Samples for Skrill Wallet

Sample of a successful Skrill Wallet request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="EUR">10.01</requested-amount>
    <account-holder>
        <!-- either set mandatoty element: payment/wallet/account-id or payment/account-
holder/email -->         <email>max.mustermann@wirecard.com</email>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <address>
                <street1>OneStoneStreet 35</street1>
                <city>MÃ¼nchen</city>
                <country>DE</country>
        </address>
  </account-holder>
  <!-- either set mandatoty element: payment/wallet/account-id or payment/account-holder/email -->
  <!-- <wallet><account-id> max.mustermann@wirecard.com</account-id></wallet> -->
  <payment-methods><payment-method name="skrill"/></payment-methods>
  <locale>de_DE</locale>
  <descriptor>Product ID:</descriptor>
  <order-number>4509334</order-number>
  <cancel-redirect-url>http://sandbox-
engine.thesolution.com/shop_urlViaRequest/cancel.html</cancel-redirect-url>
  <!-- <notifications>
        <notification                                        url="http://d-elastic-
stack0.wirecard.sys:9444/merchant/rest/success__alternative"/>
  </notifications> -->
</payment>
```

Sample of a successful Skrill Wallet response:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <transaction-id>2109370c-beea-11e2-a615-005056a9092e</transaction-id>
  <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2013-05-17T12:05:57.415Z</completion-time-stamp>
  <statuses>
    <status   code="201.0000"   description="The   resource   was   successfully   created."
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10.01</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email> max.mustermann@wirecard.com</email>
    <address>
      <street1>OneStoneStreet 35</street1>
```

```
        <city>München</city>
        <country>DE</country>
      </address>
   </account-holder>
   <order-number>4509334</order-number>
   <descriptor>Product ID:</descriptor>
   <payment-methods>
      <payment-method
url="https://www.moneybookers.com:443/app/payment.pl?sid=0c8d275a2996ef7bd8447d243fa88da0"
name="skrill"/>
   </payment-methods>
   <cancel-redirect-url>http://sandbox-
engine.thesolution.com/shop_urlViaRequest/cancel.html</cancel-redirect-url>
   <locale>de_DE</locale>
</payment>
```

Sample of a notification for a successful Skrill Wallet transaction:

```
xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
   <transaction-id>a561a25a-beea-11e2-a615-005056a9092e</transaction-id>
   <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
   <transaction-type>debit</transaction-type>
   <transaction-state>success</transaction-state>
   <completion-time-stamp>2013-05-17T12:09:38.452Z</completion-time-stamp>
   <statuses>
      <status code="200.0000" description="skrill:The request completed successfully."
severity="information"/>
   </statuses>
   <requested-amount currency="EUR">10.01</requested-amount>
   <order-number>4509334</order-number>
   <descriptor>Product ID:</descriptor>
   <payment-methods>
      <payment-method name="skrill"/>
   </payment-methods>
   <cancel-redirect-url>http://sandbox-engine.thesolution.com/shop_urlViaRequest/cancel.html</cancel-
redirect-url>
   <wallet>
      <account-id> max.mustermann@wirecard.com</account-id>
   </wallet>
   <settlement>
      <gross-amount>10.01</gross-amount>
      <currency>EUR</currency>
   </settlement>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>SzqyN8C6jtx0nxuT6t4VDfQivW8=</DigestValue>
         </Reference>
      </SignedInfo>

<SignatureValue>QcJs5pU1scWzBe7QBwY636+sdezj7ZUttkLOQxxcD4NmnUcVtFtDnZc73vjWSTL6pBsfI9v
XMGhq
qoeAZaoGpTLbEKQ5rm4Z7oKfWnijPiO4rQ4Cj4uUIi8EqCBvqCOE6NRFOkFNVWBxYOLx0wEphVxf
+xTQ3oJiw/8tbmbg7U9fUVKTWlmyJ2jMOf+1gaKxWCBTS7pUTX5qAK/o7rRgIO5DBFOhqL2eqyOk
```

```
B8V0rsYUBLflSJdeOlh4iQ1DTNhXXr5VVNEdI1Fknda+APut6gpQ3rWM+UOYypXMlMaLj2UPMTv3
PjeL7C+dNVJ+15qRfc0QL+9l/oVL5iZfO3tFKw==</SignatureValue>
    <KeyInfo>
      <X509Data>
        <X509SubjectName>CN=Manoj                    Sahu,OU=Operations,O=Wirecard                    Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>

<X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJD
QTELMAkGA1UE
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVu
dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgN
VBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAsTCk9w
ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQuyUyDeUb
QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
</payment>
```

8.3 Skrill Credit

Please see also Chapter 6 Appendix A: Payment Methods.

Skrill Credit transactions can be used in case a merchant would like to deposit money to his customer, e.g. for a dividend payout or a distribution of profit.

The transaction type within the request must be *CREDIT.*

The wallet-id of the beneficiary's is a mandatory parameter in the request but if the beneficiary is not yet registered at Skrill he has to register at Skrill within 14 days.If he doesn't do that the money will be send back from Skrill to the merchant. Although the money is not yet credited to the beneficiary's Skrill account the merchant will receive a successful response of Wirecard. After the beneficiary has opened up his Skrill account and the money was successful credited the merchant will receive a notification eMail by Skrill. The maximum amount for a Skrill credit transaction is 10.000 EUR, please be aware of that in case of a currency conversion the fees of Skrill will also be added, meaning the amount and the fees for the currency conversion may not exceed the amount of 10.000 EUR.

After the money is sent to the consumer a response with the current status of the transaction will be returned to the merchant. The merchant and the recipient of the money will additionally receive a notification eMail by Skrill.

See also XML examples for requests and response in chapter 6.3.2.

6.3.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method Skrill Credit:

Term	Man /Opt	Format	Max Size	Description
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For Skrill Credit only CREDIT is allowed in the request.
requested-amount currency	M	Numeric	18,3	This is the amount and the currency of the transaction. The amount of the decimal place is dependent of the currency. For Skrill Credit the maximum amount is 10.000 EUR.
wallet-account-id	M	Alphanumeric	50	The unique identifier of the Account Holder's Wallet Account. For Skrill Credit this is the beneficiary's username of the Skrill Wallet account.
payment-methods.payment-method-name	O	Alphanumeric	15	This is the name of the payment method. For Skrill credit the payment method is Skrill.
descriptor	M	Alphanumeric	27	Note to be included in the notification email. The following characters are allowed: umlaut, - '0-9','a-z','A-Z',' ' , '+',',','-','.'
order-number	M	Alphanumeric	64	This is the order number of the merchant. For Skrill Credit this is the subject for the notification email.

6.3.2 XML- Samples for Skrill Credit

Sample of a successful Skrill Credit request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <request-id>35dc8459-5b31-43f9-9843-0093a7655b4a</request-id>
  <transaction-type>credit</transaction-type>
  <requested-amount currency="EUR">10.01</requested-amount>
    <wallet>
            <account-id> max.mustermann@wirecard.com </account-id>
  </wallet>
  <payment-methods><payment-method name="skrill"/></payment-methods>
  <descriptor>test</descriptor>
  <order-number>4509334</order-number>
</payment>
```

Sample of a successful Skrill Credit response:

```
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-
account-      id>
        <transaction-id>d629471e-d354-11e2-ac3f-1803733b8f04</transaction-id>
        <request-id>35dc8459-5b31-43f9-9843-0093a7655b4a</request-id>
        <transaction-type>credit</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2013-06-12T13:40:45.205+02:00</completion-time-stamp>
        <statuses>
                <status code="200.0000" description="The request completed successfully."
        severity="information"/>
        </statuses>
        <requested-amount currency="EUR">10.01</requested-amount>
        <order-number>4509334</order-number>
        <descriptor>test</descriptor>
        <payment-methods>
                <payment-method name="skrill"/>
        </payment-methods>
        <wallet>
                <account-id> max.mustermann@wirecard.com</account-id>
        </wallet>
        <settlement>
                <gross-amount>10.01</gross-amount>
                <currency>EUR</currency>
        </settlement>
        </payment>
```

Sample of a failure Skrill Credit request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>02cfe8ce-1cb3-4b0e-9bd5-01fb191eaf3e</merchant-account-id>
  <request-id>0c8699bc-ae85-4647-9873-9be2596c77af</request-id>
  <transaction-type>credit</transaction-type>
  <requested-amount currency="KHR">10.01</requested-amount>
    <wallet>
            <account-id> max.mustermann@yahoo.com </account-id>
  </wallet>
  <payment-methods><payment-method name="skrill"/></payment-methods>
    <descriptor>test</descriptor>
  <order-number>4509334</order-number>
  </payment>
```

Sample of a failure Skrill Credit response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>02cfe8ce-1cb3-4b0e-9bd5-01fb191eaf3e</merchant-account-id>
<transaction-id>89efa267-d4fe-11e2-ac3f-1803733b8f04</transaction-id>
<request-id>0c8699bc-ae85-4647-9873-9be2596c77af</request-id>
<transaction-type>credit</transaction-type>
<transaction-state>failed</transaction-state>
<completion-time-stamp>2013-06-14T16:28:11.225+02:00</completion-time-stamp>
<statuses>
        <status code="400.1052" description="Currency is not supported." severity="error"/>
</statuses>
<requested-amount currency="KHR">10.01</requested-amount>
<order-number>4509334</order-number>
<descriptor>test</descriptor>
<payment-methods>
        <payment-method name="skrill"/>
</payment-methods>
<wallet>
        <account-id< max.mustermann@wirecard.com</account-id>
</wallet>
</payment>
```

8.4 PayPal

PayPal is a wallet payment method that lets consumers to make and accept payments without having to share their financial information. It is available in 190 countries worldwide, with more than 100 million active accounts. PayPal claims an average 18% increase in online sales once merchants add PayPal payment method to their online stores.

PayPal Express Checkout is a service that allows consumers to pay using a PayPal account instead of a credit card. They can make purchases without entering billing, shipping, and credit card information since that information is already stored with PayPal. PayPal expects merchants to provide details about order, so they could be shown to consumer during the checkout procedure. Also, consumer may change the shipping address within the checkout in PayPal. Therefore the shipping address together with other payment details are returned to merchant's web shop for further processing.

PayPal supports debit, authorization and capture, void, refund and authorization-only transactions (please see 8.4.3 Capture, Refund and Void for PayPal for details). If the debit or authorization request is successful, the *Forward-URL* to the PayPal Express Checkout landing page will be sent in the response.

Once Wirecard has received a notification from PayPal about the final status of the transaction, this status will be communicated to the merchant via the *Notification-URL* that is configured in the merchant account. If no *Notification-URL* is configured in the merchant account the notification will be sent via email to the merchant in case the email address has been configured. Usually, this status is received within seconds of the completion of the transaction, however, it may take up to a few days.

8.4.1 Fields for PayPal

The following elements are mandatory/optional for sending a request for the payment method PayPal.

Term	Man /Opt	Format	Max Size	Description
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Alphanumeric	30	This is the type for a transaction
requested-amount	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
payment-methods.payment-method-name	M	Alphanumeric	15	The name of the Payment Method is “paypal”.
account-holder.*	O			Various consumer details
order-detail	O	Alphanumeric	65535	Details of the order filled by the merchant.
order-number	O	Alphanumeric	64	The order number from the merchant.
shipping.*	O			Shipping details.
order-items.*	O			Basket items details.
success-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be re-directed after a successful payment. This is normally a success confirmation page on the merchant’s website.
cancel-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be re-directed after he has cancelled a payment. This is normally a page on the merchant’s website

parent-transaction-id	O	Alphanumeric	36	Transaction ID of the first transaction of a payment

8.4.2 XML Samples for PayPal

Sample of a PayPal authorization request. Note that this request leads only to a redirect URL to PayPal landing page in case of success, not completed authorization transaction. Account holder and shipping information are usually not needed, since PayPal stores them itself. Redirect and notification URLs are taken from merchant profile, and do not need to be specified in the request either.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment >
  <merchant-account-id>c57f418f-81ff-44dc-8acf-df9524c080df</merchant-account-id>
  <request-id>ba2dd7b3-a8a5-4205-98d6-6d94255a455b</request-id>
  <transaction-type>authorization</transaction-type>
  <requested-amount currency="EUR">1.03</requested-amount>
  <account-holder>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
  </account-holder>
  <shipping>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <phone>+49123123123</phone>
      <address>
            <street1>123 anystreet</street1>
            <city>Brantford</city>
            <country>CA</country>
            <postal-code>M4P1E8</postal-code>
      </address>
  </shipping>
  <order-number>48090</order-number>
  <descriptor>customerStatement 18009998888</descriptor>
  <payment-methods>
      <payment-method name="paypal"/>
  </payment-methods>
</payment>
```

Sample of a successful PayPal authorization response. The URL for redirecting consumer is provided, so they can complete authorization transaction using PayPal Express Checkout.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment >
      <merchant-account-id>c57f418f-81ff-44dc-8acf-df9524c080df</merchant-
account-id>
      <transaction-id>81119346-8cb1-11e3-a0ae-00059a3c7a00</transaction-id>
      <request-id>ba2dd7b3-a8a5-4205-98d6-6d94255a455b</request-id>
      <transaction-type>authorization</transaction-type>
      <transaction-state>success</transaction-state>
      <completion-time-stamp>2014-02-03T09:59:49.000+01:00</completion-time-
stamp>
      <statuses>
            <status code="201.0000" description="The resource was successfully
created." severity="information"/>
      </statuses>
      <requested-amount currency="EUR">1.03</requested-amount>
```

```
	<account-holder>
		<first-name>John</first-name>
		<last-name>Doe</last-name>
	</account-holder>
	<shipping>
		<first-name>John</first-name>
		<last-name>Doe</last-name>
		<phone>+49123123123</phone>
		<address>
			<street1>123 anystreet</street1>
			<city>Brantford</city>
			<country>CA</country>
			<postal-code>M4P1E8</postal-code>
		</address>
	</shipping>
	<order-number>48090</order-number>
	<descriptor>customerStatement 18009998888</descriptor>
	<payment-methods>
		<payment-method url="https://www.sandbox.paypal.com/cgi-
bin/webscr?cmd=_express-checkout&amp;useraction=commit&amp;token=EC-
7E005969CX733920H" name="paypal"/>
	</payment-methods>
</payment>
```

Sample of a notification for a successful PayPal authorization transaction (now is transaction really done).

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
	<merchant-account-id>c57f418f-81ff-44dc-8acf-df9524c080df</merchant-account-id>
	<transaction-id>93421b4b-8cb3-11e3-a0ae-00059a3c7a00</transaction-id>
	<request-id>ba2dd7b3-a8a5-4205-98d6-6d94255a455b</request-id>
	<transaction-type>authorization</transaction-type>
	<transaction-state>success</transaction-state>
	<completion-time-stamp>2014-02-03T10:14:26.000+01:00</completion-time-stamp>
	<statuses>
		<status code="201.0000" description="paypal:The resource was successfully
created." severity="information"/>
	</statuses>
	<requested-amount currency="EUR">1.030000</requested-amount>
	<ip-address>127.0.0.1</ip-address>
	<payment-methods>
		<payment-method name="paypal"/>
	</payment-methods>
	<api-id>---</api-id>
	<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
		<SignedInfo>
			<CanonicalizationMethod    Algorithm="http://www.w3.org/TR/2001/REC-
xml-c14n-20010315"/>
			<SignatureMethod   Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
			<Reference URI="">
				<Transforms>
					<Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
				</Transforms>
				<DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
				<DigestValue>Z5SmPhdnq9kZJlwUM47JRwfujHQ=</DigestValue>
			</Reference>
```

```
              </SignedInfo>
       <SignatureValue>hA+Ppr5ZgyUNuJsjq8Ry76F3YM2iEcVYNMd5FM24pXwSLPFOSWuush/XjoXw+nzLS
EhCe8S9dYbQ
peC5Y7/7xoYIYPi2Pwp85JD4bZmKPB+K5GVGo2hMsjMICWDdsQle3h2Kf5wgND+puGWcNcE5VRb4
BTZOMhn/5QiSp9VAeBGGlAoLj10302JF6ykt1kZdB1RJX6DGyXNOrUjSHKUhSSj7+6ZJrNoT28+E
YTl/RUX4CVD2R724zFMm4NrgNML7znNUkdDA/e4SwsBbXJ2gKX8LskIII9EeKTxahi2su3G+HGgl
YSGjTwt+EiyhdKj9e2LHbW7vjpOgfBRV6bf36w==</SignatureValue>
              <KeyInfo>
                     <X509Data>
                            <X509SubjectName>CN=Manoj       Sahu,OU=Operations,O=Wirecard
Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>
       <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJD
QTELMAkGA1UE
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVu
dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAsTCk9w
ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQuyUyDeUb
QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
                     </X509Data>
              </KeyInfo>
       </Signature>
</payment>
```

8.4.3 Capture, Refund and Void for PayPal

CAPTURE-AUTHORIZATION | Available only on successful AUTHORIZATION which wasn't voided nor fully captured yet. Partial and multiple capture is allowed, until the authorization limit is not exceeded.

VOID-AUTHORIZATION | Available only on successful AUTHORIZATION which wasn't voided nor captured yet. Authorization is voided as a whole.

REFUND-DEBIT | Available on successful DEBIT. Partial and multiple refund is allowed, until the original amount is not exceeded.

REFUND-CAPTURE | Available on successful CAPTURE-AUTHORIZATION. Partial and multiple refund is allowed, until the original amount is not exceeded.

8.4.4 Reference transactions for PayPal

To submit a recurring transaction, the merchant must submit a request with the transaction type DEBIT, AUTHORIZATION or AUTHORIZATION-ONLY including the PERIODIC TYPE element with type "recurring" or "installment". Additionally, the sub-element SEQUENCE TYPE with one of the following sequences must be submitted:

- first: The first transaction in a series of recurring transactions
- recurring: A transaction that is part of a series of recurring transactions.

- final: The final transaction in a series of recurring transactions. A payment with this sequence type completes a chain of recurring payments.

AUTHORIZATION-ONLY transaction can be only the first one in the series, and the amount is always zero. It therefore only effectively creates a valid Billing Agreement ID with PayPal, without really charging consumer at that occasion (unlike AUTHORIZATION or DEBIT).

The PERIODIC TYPE element is differentiation for the merchant's use only, and which of the two periodic types is chosen depends on the merchant's business model. 'Installment': one in a set that completes a financial transaction and 'recurring': one in a set that occurs repeatedly, such as a subscription. Both 'recurring' and 'installment' transactions will be processed the same way.

“Recurring” and “final” transactions need to reference (parent-transaction ID) a valid “first” transaction. After a transaction with sequence type “final” has been submitted, it is not possible to submit another “recurring” or “final” transaction for this series. Only one periodic type may be used for the complete series of recurring payments. This means that the periodic type in a subsequent transaction must match the periodic type sent in the “first” request.

The following fields are relevant for a recurring transactions:

Term	Man/Opt Req/Resp	Format	Max Size	Description
parent-transaction-id	M* (req) *mandatory for “recurring” and “final” txs	Alphanumeric	36	Transaction ID of the first transaction in the series
periodic.periodic-type	M (req)	Alpha	9	Indicates how and why a Payment occurs more than once. Only two possible values: 'recurring' or ‘installment’
periodic.periodic-type.sequence-type	M (req)	Alpha	9	Indicates the sequence of the recurring transaction. Possible values: ‘first’, ‘recurring’ or ‘final’
provider-transaction-reference-id	O	Alphanumeric		Billing agreement ID from PayPal returned in response.

8.4.5 Cross merchant account references

The default security setting for all types of reference transactions using the field payment/parent-transaction-id allows the references only to the transactions of the same merchant account. However it is possible to activate cross referencing for a merchant account. With this option activated a merchant account can reference a transaction of another merchant account. The only requirement is that the processing user used for the authentication of the payment request has rights on the merchant account of the referenced transaction.
Please note: this merchant configuration parameter affects all transaction types. Other kinds of reference transactions such as refund or void can use cross references as well. Once activated it is enabled for all payment methods configured for a merchant account even though not all payment providers/acquirers might support it.

PayPal supports cross references natively by configuring a special shared customer account which can be used by several merchants. Therefore the billing agreement ID can also be shared among several merchants. The cross references must be solely enabled for the corresponding merchant accounts of the processing user. The recurring transactions process in cross-reference mode remains the same as described in **8.4.4 Reference transactions for PayPal.**

8.4.6 Basket items for PayPal

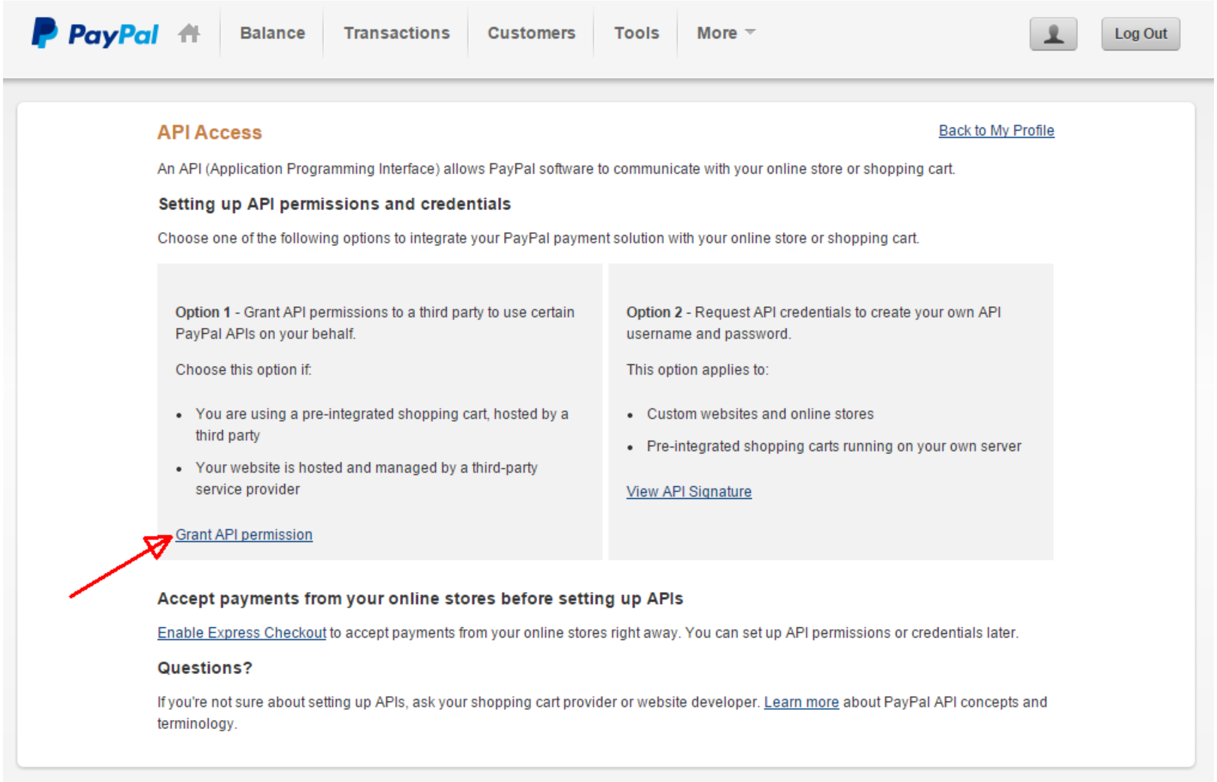
Basket items details might be sent along with debit and authorization requests. The detailed items of the shopping basket will be displayed in PayPal during the checkout, and later stored as a part of payment details in PayPal. They can be obtained from Wirecard Processing API later as part of transaction details as well. The following fields are maintained:

Term	Man/ Opt Req/ Resp	Format	Max Size	Description
order-items.order-item.name	O	Alphanumeric		Name of the item in the basket.
order-items.order-item.description	O	Alphanumeric		Longer item description.
order-items.order-item.article-number	O	Alphanumeric		EAN or other article identifier for merchant.
order-items.order-item.amount	O	Number		Item’s price per unit.
order-items.order-item.tax-amount	O	Number		Item’s tax per unit.
order-items.order-item.quantity	O	Number		Total count of items in the order.

8.4.7 Granting 3rd party permission on PayPal

In order to submit payments on merchant’s behalf, merchant needs to grant 3rd party API permission to Wirecard on PayPal website.

1. Navigate to “Account settings” → “My selling tools” → “API Access” and select “Grant API permission”.



API Access

Back to My Profile

An API (Application Programming Interface) allows PayPal software to communicate with your online store or shopping cart.

Setting up API permissions and credentials

Choose one of the following options to integrate your PayPal payment solution with your online store or shopping cart.

Option 1 - Grant API permissions to a third party to use certain PayPal APIs on your behalf.

Choose this option if:

- You are using a pre-integrated shopping cart, hosted by a third party
- Your website is hosted and managed by a third-party service provider

Grant API permission

Option 2 - Request API credentials to create your own API username and password.

This option applies to:

- Custom websites and online stores
- Pre-integrated shopping carts running on your own server

View API Signature

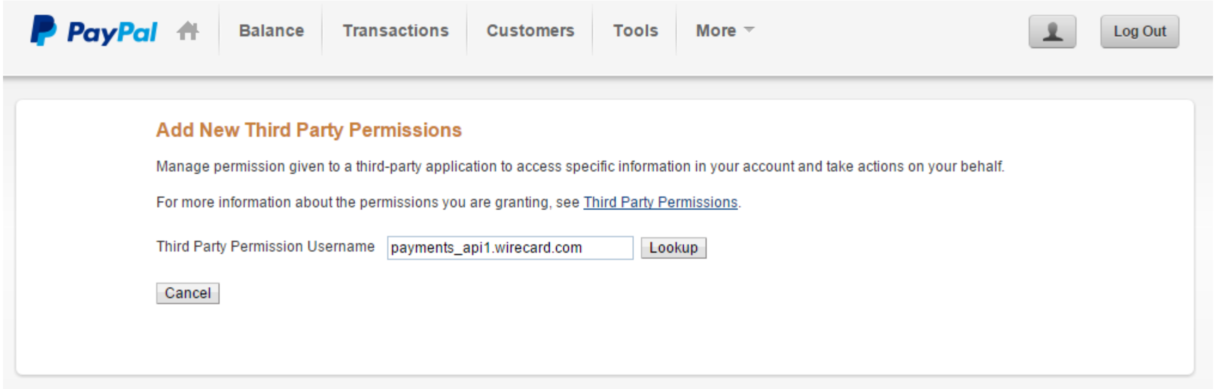
Accept payments from your online stores before setting up APIs

Enable Express Checkout to accept payments from your online stores right away. You can set up API permissions or credentials later.

Questions?

If you're not sure about setting up APIs, ask your shopping cart provider or website developer. Learn more about PayPal API concepts and terminology.

2. Enter “payments_api1.wirecard.com” as the “Third Party Permission Username” and click on “Lookup”.



Add New Third Party Permissions

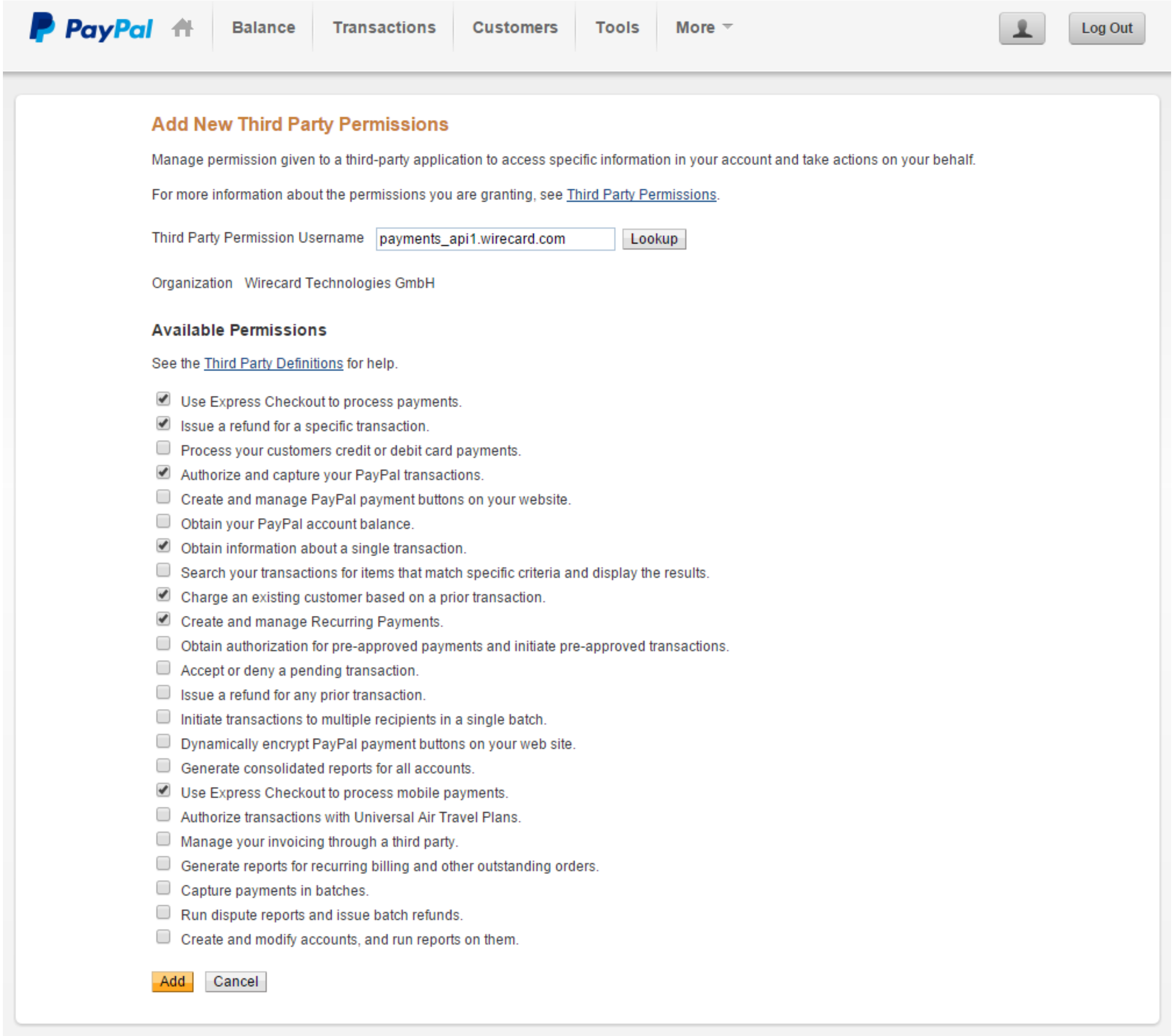
Manage permission given to a third-party application to access specific information in your account and take actions on your behalf.

For more information about the permissions you are granting, see Third Party Permissions.

Third Party Permission Username: payments_api1.wirecard.com [Lookup]

[Cancel]

3. Select from the list of “Available Permissions” following items:
   - “Use Express Checkout to process payments.”
   - “Issue a refund for a specific transaction.”
   - “Authorize and capture your PayPal transactions.”
   - “Obtain information about a single transaction.”
   - “Charge an existing customer based on a prior transaction.”
   - “Create and manage Recurring Payments.”
   - “Use Express Checkout to process mobile payments.”



Add New Third Party Permissions

Manage permission given to a third-party application to access specific information in your account and take actions on your behalf.

For more information about the permissions you are granting, see Third Party Permissions.

Third Party Permission Username payments_api1.wirecard.com Lookup

Organization Wirecard Technologies GmbH

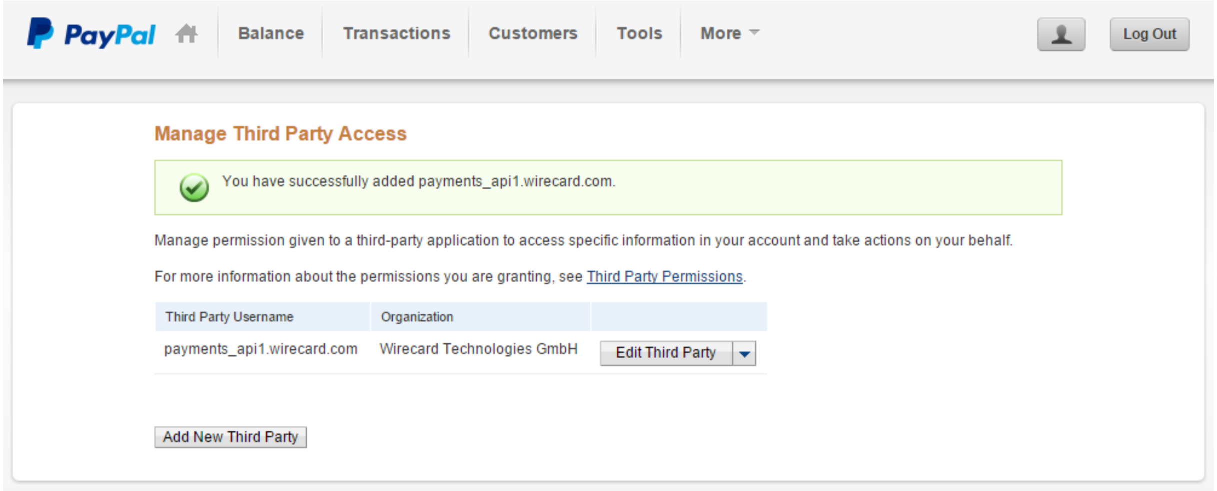
Available Permissions

See the Third Party Definitions for help.

- ☑ Use Express Checkout to process payments.
- ☑ Issue a refund for a specific transaction.
- □ Process your customers credit or debit card payments.
- ☑ Authorize and capture your PayPal transactions.
- □ Create and manage PayPal payment buttons on your website.
- □ Obtain your PayPal account balance.
- ☑ Obtain information about a single transaction.
- □ Search your transactions for items that match specific criteria and display the results.
- ☑ Charge an existing customer based on a prior transaction.
- ☑ Create and manage Recurring Payments.
- □ Obtain authorization for pre-approved payments and initiate pre-approved transactions.
- □ Accept or deny a pending transaction.
- □ Issue a refund for any prior transaction.
- □ Initiate transactions to multiple recipients in a single batch.
- □ Dynamically encrypt PayPal payment buttons on your web site.
- □ Generate consolidated reports for all accounts.
- ☑ Use Express Checkout to process mobile payments.
- □ Authorize transactions with Universal Air Travel Plans.
- □ Manage your invoicing through a third party.
- □ Generate reports for recurring billing and other outstanding orders.
- □ Capture payments in batches.
- □ Run dispute reports and issue batch refunds.
- □ Create and modify accounts, and run reports on them.

Add Cancel

4. Check that you was able to successfully add the permissions.



Manage Third Party Access

You have successfully added payments_api1.wirecard.com.

Manage permission given to a third-party application to access specific information in your account and take actions on your behalf.

For more information about the permissions you are granting, see Third Party Permissions.

Third Party Username	Organization	
payments_api1.wirecard.com	Wirecard Technologies GmbH	Edit Third Party

Add New Third Party

Finally, Wirecard needs to know merchant’s PayPal registration e-mail address to submit payments.

8.5 ePlatby

ePlatby (also known as ePlatby for eKonto or just eKonto in the past) is an online bank transfer solution offered by Raiffeisenbank in the Czech Republic. Upon selecting ePlatby as their preferred payment method on the merchant's website, consumers are transferred to their bank's website where they login using their online banking credentials and confirm the payment with a TAN.

ePlatby supports only debit transaction (please see **Error! Reference source not found.** for details). If the DEBIT request is successful, the *Forward-URL* to the ePlatby landing page will be sent in the response to redirect the consumer for payment approval. Merchant is informed about the result of payment by a notification afterwards.

The final status of the transaction will be communicated to the merchant via the *Notification-URL* that is configured in the merchant account or was passed along with the request. If no *Notification-URL* is configured in the merchant account the notification will be sent via email to the merchant in case the email address has been configured. In case the notification hasn't received within 40 minutes, the merchant can request the status of the transaction by sending a "Retrieve Transaction by Transaction ID" (see Chapter 0) or "Retrieve Transaction by Request ID" (see Chapter 5.4).

8.5.1 Fields for ePlatby

The following elements are mandatory/optional for sending a request for the payment method ePlatby.

Term	Man /Opt	Format	Max Size	Description
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Alphanumeric	30	This is the type for a transaction
requested-amount	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
payment-methods.payment-method-name	M	Alphanumeric	15	The name of the Payment Method is "eplatby".
order-description	O	Alphanumeric	65535	Description for merchant which will be stated in the record of transaction.
order-number	O	Alphanumeric	64	The order number from the merchant, which will be used as variable symbol of the payment.

success-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be re-directed after a successful payment. This is normally a success confirmation page on the merchant’s website.
cancel-redirect-url	O	Alphanumeric	256	The URL to which the consumer will be re-directed after he has cancelled a payment. This is normally a page on the merchant’s website

8.5.2 XML Samples for ePlatby

Sample of an ePlatby debit request. Note that this request leads only to a redirect URL to ePlatby landing page in case of success, not completed debit transaction. Redirect and notification URLs are taken from merchant profile, and do not need to be specified in the request.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment>
    <merchant-account-id>36b9dd7c-88a5-4cc9-b25c-171cccb86bcf</merchant-account-
  id>
    <request-id>a04464bf-0f04-4c2b-a6c0-bd39b72f4bb7</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="CZK">1.01</requested-amount>
    <order-number>22827</order-number>
  <order-description>Order 22827 from thebookshop.cz</order-description>
    <payment-methods>
        <payment-method name="eplatby"/>
    </payment-methods>
</payment>
```

Sample of a successful ePlatby debit response. The URL for redirecting consumer is provided, so they can complete authorization transaction using ePlatby site.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment>
    <merchant-account-id>36b9dd7c-88a5-4cc9-b25c-171cccb86bcf</merchant-account-id>
    <transaction-id>57d70b64-be4b-11e3-be85-0050b66a327a</transaction-id>
    <request-id>a04464bf-0f04-4c2b-a6c0-bd39b72f4bb7</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2014-04-07T13:54:18.000+02:00</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully created."
severity="information"/>
    </statuses>
    <requested-amount currency="CZK">1.01</requested-amount>
    <order-number>22827</order-number>
    <payment-methods>
        <payment-method
```

```
url="https://klient2.rb.cz/test_shop/owa/shop.payment?URLfail=https%3A%2F%2Fsandbox-
engine.thesolution.com%2Fshop%2Fcancel.html&amp;URLsuccess=http%3A%2F%2Flocalhost%3A808
0%2Fengine%2Fnotification%2Fekonto&amp;amount=1.01&amp;creditaccount=1212121001&amp;cre
ditbank=5500&amp;credittext=&amp;curcode=CZK&amp;otherparams=payment.transaction-
id%3D57d70b64-be4b-11e3-be85-0050b66a327a&amp;shopname=EKONTO&amp;varsymbol=22827"
                name="eplatby"/>
    </payment-methods>
</payment>
```

Sample of a notification for a successful ePlatby authorization transaction (now is transaction really done).

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36b9dd7c-88a5-4cc9-b25c-171cccb86bcf</merchant-account-id>
    <transaction-id>5f99f7df-be4b-11e3-be85-0050b66a327a</transaction-id>
    <request-id>a04464bf-0f04-4c2b-a6c0-bd39b72f4bb7</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2014-04-07T13:54:31.000+02:00</completion-time-stamp>
    <statuses>
        <status  code="201.0000"  description="ekonto:The  resource  was  successfully
created." severity="information"/>
    </statuses>
    <requested-amount currency="CZK">1.010000</requested-amount>
    <order-number>22827</order-number>
    <payment-methods>
        <payment-method name="eplatby"/>
    </payment-methods>
    <api-id>---</api-id>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod   Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <Reference URI="">
                <Transforms>
                    <Transform   Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <DigestValue>iZBfzp82lilnWdJNnoNxQcqarCI=</DigestValue>
            </Reference>
        </SignedInfo>
       <SignatureValue>Al5px4qOae6Fp3DZm+9SHv6wOpeS4AIC0l+LmCe19JdwmquxeGrh+s6zxz7aewmw
       zTvO5KSHrbYdOV7K8fKINGYLZv/JNvh+URUcvtUxoBnHWtpe5aKDn8I8TgKNp0y3kkkewn5GcknavCie
       9h1OB5YQ50qSWVS+Z9Fm39tD3jlJg9b1G5xhWwRkF5YODoKr83AQFyL11DVWCWrZzVZZEGtsmh0lrKgy
       xSc1XzhSKLhvYYvXUfw9X109GaLt7zfP0Xw5WFuDAd30O408hTeExL2Hp5qi9H+0ztlsJSi0+McJ6+vm
       xmDznT+cXFDG6QvxdALJ9KMc28K5K+iXVBe3pA==
        </SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509SubjectName>CN=Manoj      Sahu,OU=Operations,O=Wirecard      Elastic
Payments,L=Toronto,ST=ON,C=CA
                </X509SubjectName>
                <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQ
                YDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpc
                mVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMT
                Ck1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQ
                GEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYX
                JkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hb
                m9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCc8rTt4N5fNeVzlsRg
                OKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCNetBvJ9rgNc0Cxrk/euMj
                3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5HaJV4n5cXJQ/uhAEYU3
                EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW7VgKD2pqu6BRysq
                UVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQuyUyDeUbQmhVQ
                KL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB
                ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXo
                yovZrmk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBf
                IS5TJ54e7m02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8
                KCHpfw2j8G+dDgUjlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpn
                sZ9NFfoJNdG13AzYwDP3x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=
                </X509Certificate>
            </X509Data>
        </KeyInfo>
    </Signature>
</payment>
```

8.6 Guaranteed Invoice, Direct Debit and Installments

Guaranteed is online payment method offering payments by invoice and direct debit, popular among German online shoppers. Guaranteed Invoice, Direct Debit and Installments takes over the risk, paying the full amount to merchant and collecting from the shopper later.

Wirecard Payment Gateway supports payments by both methods. Guaranteed Invoice (ratepay-invoice) lets the consumer to pay by invoice, only after the goods were delivered. Guaranteed Direct Debit (ratepay-elv) works in similar way, consumer is charged only after the delivery.

Guaranteed Invoice, Direct Debit and Installments implementation supports authorization and capture, void, refund and credit. Every Guaranteed transaction must contain a list of remaining items in the currently valid shopping basket. It does not contain the delta to a previously sent list. Please see chapters 8.6.3 and 8.6.4 for details.

The response for each authorization request returns an individual string in the field DESCRIPTOR. The descriptor must be printed on all invoice documents and be saved by the shop. The descriptor (a.k.a. “narrative”) must be used by the customer on bank transfers (German: “Verwendungszweck”).

The capture transaction has to follow a successful authorization on the same day the goods are dispatched. The merchant can do partials captures, as are parts of the ordered goods delivered.

8.6.1 Fields for Guaranteed Invoice, Direct Debit and Installments

The following elements are mandatory/optional for sending a request for the payment method Guaranteed.

Term	Man /Opt	Format	Max Size	Description
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**

transaction-type	M	Alphanumeric	30	This is the type for a transaction: “authorization”, “capture”, “void-authorization”, “refund-capture” and “credit” are supported for Guaranteed Invoice, Direct Debit and Installments.
requested-amount	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
payment-methods.payment-method-name	M	Alphanumeric	15	The name of the Payment Method is “ratepay-invoice” or “ratepay-elv”.
account-holder.*	M			Various consumer details including the address.
bank-account.*	M/O			Consumers bank account details. Mandatory for Guaranteed Direct Debit only.
order-items.*	M			Basket items details.
order-number	M	Alphanumeric	64	The order number from the merchant.
shipping.*	O			Shipping details.
order-detail	O	Alphanumeric	65535	Details of the order filled by the merchant.
consumer-id	O	Alphanumeric		Merchant’s identifier for the consumer.
custom-field “invoice-id”	O	Alphanumeric		Merchant’s invoice identifier.

Bank account details consist either of ACCOUNT-OWNER, BANK-NAME, ACCOUNT-NUMBER and BANK-CODE; or ACCOUNT-OWNER, BANK-NAME, IBAN and BIC for SEPA standard.

Account holder details must cover FIRST-NAME, LAST-NAME, EMAIL, PHONE, DATE-OF-BIRTH and ADDRESS. GENDER is optional. Address within the account holder details must include STREET1, CITY, COUNTRY and POSTAL-CODE.

8.6.2 XML Samples for Guaranteed Invoice, Direct Debit and Installments

Sample of a Guaranteed Direct Debit authorization request.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>de4853bb-b0ce-4b85-bc18-040a11400c8a</merchant-account-id>
  <request-id>8cb0fb78-abad-4131-91bb-7b6771534b5a</request-id>
  <transaction-type>authorization</transaction-type>
  <payment-methods>
      <payment-method name="ratepay-elv"/>
  </payment-methods>
  <requested-amount currency="EUR">55</requested-amount>
  <order-number>27783</order-number>
  <account-holder>
      <first-name>Hans-Jürgen</first-name>
      <last-name>Wischnewski</last-name>
      <email>john.doe@test.com</email>
      <phone>03018425165</phone>
      <date-of-birth>07-12-1977</date-of-birth>
      <address>
             <street1>Müllerstraße 137</street1>
             <city>Berlin</city>
             <state>Berlin</state>
             <country>DE</country>
             <postal-code>13353</postal-code>
      </address>
  </account-holder>
  <bank-account>
      <account-number>4989123456</account-number>
       <bank-code>20320500</bank-code>
      <bank-name>Danske Bank Hamburg</bank-name>
  </bank-account>
  <order-items>
      <order-item>
             <name>Shirt rot</name>
             <article-number>11</article-number>
             <amount currency="EUR">10.00</amount>
             <tax-rate>19</tax-rate>
             <quantity>2</quantity>
      </order-item>
      <order-item>
             <name>Hose blau</name>
             <article-number>21</article-number>
             <amount currency="EUR">20.00</amount>
             <tax-rate>19</tax-rate>
             <quantity>1</quantity>
      </order-item>
      <order-item>
             <name>Hut grau</name>
             <article-number>31</article-number>
             <amount currency="EUR">15.00</amount>
             <tax-rate>19</tax-rate>
             <quantity>1</quantity>
      </order-item>
  </order-items>
  <custom-fields>
      <custom-field field-name="invoice" field-value=""/>
  </custom-fields>
</payment>
```

Sample of a successful Guaranteed Direct Debit authorization response.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>de4853bb-b0ce-4b85-bc18-040a11400c8a</merchant-account-id>
   <transaction-id>ad7bf95e-4486-11e4-96d7-005056a9092e</transaction-id>
   <request-id>8cb0fb78-abad-4131-91bb-7b6771534b5a</request-id>
   <transaction-type>authorization</transaction-type>
   <transaction-state>success</transaction-state>
   <completion-time-stamp>2014-09-25T07:36:38.000Z</completion-time-stamp>
   <statuses>
```

```
      <status code="201.0000" description="The resource was successfully created."
severity="information"/>
   </statuses>
   <requested-amount currency="EUR">55</requested-amount>
   <account-holder>
      <first-name>Hans-Jürgen</first-name>
      <last-name>Wischnewski</last-name>
      <email>john.doe@test.com</email>
      <date-of-birth>1977-12-07</date-of-birth>
      <phone>03018425165</phone>
      <address>
         <street1>Müllerstraße 137</street1>
         <city>Berlin</city>
         <state>Berlin</state>
         <country>DE</country>
         <postal-code>13353</postal-code>
      </address>
   </account-holder>
   <order-number>27783</order-number>
   <order-items>
      <order-item>
         <name>Shirt rot</name>
         <article-number>11</article-number>
         <amount currency="EUR">10.00</amount>
         <tax-rate>19</tax-rate>
         <quantity>2</quantity>
      </order-item>
      <order-item>
         <name>Hose blau</name>
         <article-number>21</article-number>
         <amount currency="EUR">20.00</amount>
         <tax-rate>19</tax-rate>
         <quantity>1</quantity>
      </order-item>
      <order-item>
         <name>Hut grau</name>
         <article-number>31</article-number>
         <amount currency="EUR">15.00</amount>
         <tax-rate>19</tax-rate>
         <quantity>1</quantity>
      </order-item>
   </order-items>
   <custom-fields>
      <custom-field field-name="invoice" field-value=""/>
   </custom-fields>
   <descriptor>DG0415745D4</descriptor>
   <payment-methods>
      <payment-method name="ratepay-elv"/>
   </payment-methods>
   <bank-account>
      <bank-name>Danske Bank Hamburg</bank-name>
      <account-number>4989123456</account-number>
      <bank-code>20320500</bank-code>
   </bank-account>
</payment>
```

8.6.3 Capture, Refund, Void and Credit for Guaranteed

Basket items must be sent along with every Guaranteed request. Content of the shopping basket depends on location of the items. It can be differed between 3 locations: The (distribution) storage of the merchant, the delivery (items are „on the road") and the home of the consumer. The shopping basket always contains the items that are remaining after the action is done.

AUTHORIZATION | Reserves funds at the moment when buyer makes an order. Basket items list all ordered items. Amount of authorization must match the value of the items in the basket.

CAPTURE-AUTHORIZATION | Used to capture the reserved funds at the moment when the some or all goods are physically delivered to the consumer. Available only on successful AUTHORIZATION which wasn't voided nor fully captured yet. Partial (and multiple) capture is allowed, until the authorization limit is not exceeded. Basket items only lists the items that were delivered to the consumer in this batch, not the ones that were deliver before or are just waiting for a delivery. Amount of capture must be the value of the items in the basket (delivered).

VOID-AUTHORIZATION | Used when consumer wishes to cancel some or all (not yet shipped) items from the order. Available only on successful AUTHORIZATION which wasn't voided nor captured yet. Partial and multiple void is allowed, until all items from the order are not canceled. Basket items only list those that stay still ordered and not delivered to consumer yet, not those which are being canceled from the order.

REFUND-CAPTURE | Used when consumer wishes to return some or all already delivered goods. Available on successful CAPTURE-AUTHORIZATION. Partial and multiple refund is allowed, until all delivered items are not returned. Basket items list those items which consumer still kept, not the ones which are being returned to the merchant.

CREDIT | Used when merchant wishes to grant a discount on existing order. Items from this order might be already delivered to the consumer. Available on successful AUTHORIZATION. Multiple credit is allowed, but the overall sum of credits cannot exceeded the value of delivered goods to the consumer. Credit is just another basket item added to the order to Guaranteed, therefore basket must include one item representing the discount in this case.

8.6.4 Basket items for Guaranteed Invoice, Direct Debit and Installments

Basket items details must be sent along with every request. Content of the shopping basket depends on location of the items. It can be differed between 3 locations: The (distribution) storage of the merchant, the delivery (items are „on the road") and the home of the consumer. The shopping basket always contains the items that are remaining after the action is done.

The detailed items of the shopping basket will be stored as a part of payment details on provider and they can be obtained from Wirecard Processing API later as part of transaction details. The following fields are maintained:

Term	Man/ Opt Req/ Resp	Format	Max Size	Description
order-items.order-item.name	M	Alphanumeric		Name of the item in the basket.
order-items.order-item.article-number	M	Alphanumeric		EAN or other article identifier for merchant.
order-items.order-item.amount	M	Number		Item's price per unit.
order-items.order-item.tax-rate	M	Number		Item's tax rate per unit.
order-items.order-item.quantity	M	Number		Total count of items in the order.

8.7 MasterPass

MasterPass is a simple digital wallet for faster safe shopping, allowing to make purchases without entering shipping, and credit card information. However, MasterPass only stores credit card and shipping details for buyers, and doesn't provide any payment method as such.

Buyer is redirected to MasterPass site to finish the checkout. Buyer selects the card to pay with and may also select the shipping address for the order. Wirecard Payment Gateway retrieves the card details from MasterPass and automatically submits the card payment to an acquirer. Every successful MasterPass payment request will therefore end up with two distinct transactions in the database: a MasterPass debit and credit card purchase.

Once Wirecard has received a notification from MasterPass about the final status of the transaction, this status will be communicated to the merchant via the *Notification-URL* that is configured in the merchant account. If no *Notification-URL* is configured in the merchant account the notification will be sent via email to the merchant in case the email address has been configured. Usually, this status is received within seconds of the completion of the transaction, however, it may take up to a few days.

8.7.1 Fields for MasterPass

The following elements are mandatory/optional for sending a request for the payment method MasterPass.

Term	Man /Opt	Format	Max Size	Description
merchant-account-id	M	Alphanumeric	36	Unique identifier for a merchant account
request-id	M	Alphanumeric	150	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Alphanumeric	30	This is the type for a transaction
requested-amount	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
payment-methods.payment-method-name	M	Alphanumeric	15	The name of the Payment Method is “ratepay-invoice” or “ratepay-elv”.
shipping.*	O			Shipping details.

8.7.2 XML Samples for MasterPass

Sample of a MasterPass debit request.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>ff1434a2-b358-41ea-bc84-a81d0cce4678</merchant-account-id>
  <request-id>8d72c32f-9a3f-49bf-9388-54e3d3b34098</request-id>
  <transaction-type>debit</transaction-type>
  <payment-methods>
```

```
      <payment-method name="masterpass" />
   </payment-methods>
   <requested-amount currency="EUR">1.00</requested-amount>
</payment>
```

Sample of a successful MasterPass debit response with redirect-url.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>ff1434a2-b358-41ea-bc84-a81d0cce4678</merchant-account-id>
    <transaction-id>dbfc7274-5932-11e4-abc6-0050b6685bff</transaction-id>
    <request-id>8d72c32f-9a3f-49bf-9388-54e3d3b34098</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2014-10-21T16:59:34.000+02:00</completion-time-stamp>
    <statuses>
       <status code="201.0000" description="The resource was successfully created."
       severity="information"/>
    </statuses>
    <requested-amount currency="EUR">1.00</requested-amount>
    <payment-methods>
       <payment-method
       url="http://localhost:8080/engine/hpp/masterpassLightbox?oauth_token=09aee047d1a1
       29f71451a50e218932f6562d366f&amp;acceptable_cards=master,visa&amp;checkout_identi
       fier=a466w4xi73dtqhzd3btie1i06e0d3bcfl7&amp;version=v6&amp;callback=http%3A%2F%2F
       localhost%3A8080%2Fengine%2Fnotification%2Fmasterpass%3Fpayment.transaction-
       id%3Ddbfc7274-5932-11e4-abc6-0050b6685bff%26payment.transaction-
       type%3Ddebit&amp;lightboxurl=https%3A%2F%2Fsandbox.masterpass.com%2Flightbox%2FSw
       itch%2Fintegration%2FMasterPass.client.js" name="masterpass"/>
    </payment-methods>
</payment>
```

8.7.3 Void and Refund for MasterPass

Successful MasterPass debit request automatically submits regular credit card purchase transaction. All usual follow-up card transaction are available, so purchase can be voided, refunded or reversed referencing to original transaction.

8.7.4 Shipping address for MasterPass

Buyer may select shipping address stored within MasterPass e-wallet during the checkout. The shipping details will be send in notification to merchant, and they can be obtained from Wirecard Processing API later as part of transaction details.

8.7.5 Merchant onboarding for MasterPass

Merchant onboarding for MasterPass is rather complicated process requiring merchant’s and Wirecard’s active participation. Please see chapter “Service Provider Merchant by Merchant Onboarding - Steps”, steps 3 to 10 in detailed document by MasterPass.

8.8 Carrier Billing

Please see also Appendix A: Payment Methods

The payment method within the request must be *carrier-billing*, the transaction type *debit.*
If the request is successful the *Forward-URL* to the Landing-Page of the carrier billing provider will be sent in the response. The notification allows a merchant to receive the final status of a payment as soon as Wirecard receives it from the carrier billing provider. Usually, this status is received within seconds of the completion of the transaction, however, may take up to a few days depending on the respective mobile provider.

Once Wirecard has received a notification from the carrier billing provider about the final status of the transaction this status will be communicated to the merchant via the NotificationURL that is configured in the merchant account or provided with the initial payment transaction request. If no notificationURL is provided with the request and also not configured in the merchant account, the notification will be sent via Email to the merchant in case the Email address has been configured. In case the notification hasn't been received the merchant can request the status of the transaction by sending a "Retrieve Transaction by Transaction ID" (see Chapter 0) or "Retrieve Transaction by Request ID" (see Chapter 5.4).

8.8.1 Fields for carrier billing

The following elements are mandatory/optional for sending a request/receiving a response for the payment method carrier-billing:

Term	Man /Opt	Request/ Response	Format	Max Size	Description
merchant-account-id	M	Req, Resp	Alphanumeric	36	Unique identifier for a merchant account.
transaction-id	M	Resp	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	M	Req, Resp	Alphanumeric	64	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Req, Resp	Alphanumeric	30	This is the type for a transaction. For carrier billing only DEBIT is allowed in the initial request.
state	M	Resp	Alphanumeric	12	The payment transaction state. For carrier billing can only be success, failed or in-progress.
completion-time-stamp	M	Resp	Datetime		The completion timestamp of the transaction processing.
statuses.status@code	M	Resp	Alphanumeric	12	This is the code of the status of a transaction

statuses.status@description	M	Resp	Alphanumeric	256	This is the description to the status code of a transaction
statuses.status@severity	M	Resp	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
requested-amount	M	Req, Resp	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency. The maximum amount is highly dependent on the country and mobile network operator. Currently the maximal allowed amount is 30 EUR.
requested-amount@currency	M	Req, Resp	Alphanumeric	3	The ISO code of the payment currency. Currently only EUR is supported.
parent-transaction-id	O	Resp	Alphanumeric	36	Transaction ID of the first transaction in the series
account-holder.phone	O	Req, Resp	Alphanumeric	30	The phone number of the end-customer (MSISDN) intended to be used for payment
order-detail	O	Req, Resp	Alphanumeric	20	Additional description of the provided product or service.
descriptor	M	Req, Resp	Alphanumeric	40	Description of the provided product or service. It will appear on the checkout web page and SMS texts and may also appear on the end-customers billing invoice from the mobile operator depending on the country and operator.
notifications.notification@url	O	Req, Resp	Alphanumeric	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
payment-methods.payment-method-name@name	M	Req, Resp	Alphanumeric	15	This is the name of the payment method that that is chosen from the end-consumer. Currently only carrier-billing is supported.

payment-methods.payment-method-name@url	O	Resp	Alphanumeric	256	The forward URL to the carrier billing provider checkout page. The end-consumer must be redirected to this URL in order to be able to complete the payment.
consumer-id	O	Req, Resp	Alphanumeric	50	An id of the end-consumer in the merchant’s application. E.g. account name, gamer alias, login username.
processing-redirect-url	M	Req, Resp	Alphanumeric	256	The URL to which the consumer will be redirected after he has fulfilled his payment but the status is not known yet and the processing may take a long time. This is normally a page on the merchant’s website.
cancel-redirect-url	M	Req, Resp	Alphanumeric	256	The URL to which the consumer will be re-directed after he has cancelled a payment. This is normally a page on the merchant’s website
fail-redirect-url	M	Req, Resp	Alphanumeric	256	The URL to which the consumer will be re-directed after the payment has failed for some reason. This is normally a page on the merchant’s website
success-redirect-url	M	Req, Resp	Alphanumeric	256	The URL to which the consumer will be re-directed after a successful payment. This is normally a success confirmation page on the merchant’s website.
locale	M	Req, Resp	Alphanumeric	6	ISO code of the language. Can be sent in the format <language> or in the format <language_country>.
account-holder.address.country	M	Req, Resp	Alphanumeric	3	The ISO code of the country used for the mobile payment. It influences the language of the checkout page and usage of the mobile operators. Currently only DE is supported.

8.8.2 Refund for Carrier Billing

URL	https://hostname/engine/rest/payments
Request Formats	XML
Response Formats	XML
Request Methods	POST
Transaction Type	refund-debit

REFUND-DEBIT | Available on successful DEBIT. Partial and multiple refund are not allowed. Only the original transaction amount can be refunded.

8.8.2.1 Refund fields

The following elements are mandatory/optional for sending a refund request/receiving a response for the payment method carrier-billing:

Term	Man /Opt	Request/ Response	Format	Max Size	Description
merchant-account-id	M	Req, Resp	Alphanumeric	36	Unique identifier for a merchant account. For refunds it must be the same merchant account id as for the debit transaction to be refunded.
transaction-id	M	Resp	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	M	Req, Resp	Alphanumeric	64	This is the identification number of the request. **It has to be unique for each request.**
transaction-type	M	Req, Resp	Alphanumeric	30	This is the type for a transaction - refund-debit.
state	M	Resp	Alphanumeric	12	The refund transaction state. For carrier billing can only be success, failed or in-progress.
completion-time-stamp	M	Resp	Datetime		The completion timestamp of the refund transaction processing.
statuses.status@code	M	Resp	Alphanumeric	12	This is the code of the status of a refund transaction
statuses.status@description	M	Resp	Alphanumeric	256	This is the description to the status code of a refund transaction
statuses.status@severity	M	Resp	Alphanumeric	20	This field gives information if a status is a warning, an error or an information

parent-transaction-id	O	Resp	Alphanumeric	36	Transaction ID of the initial debit transaction to be refunded
descriptor	M	Req, Resp	Alphanumeric	40	The reason for refunding, a human readable comment.
payment-methods.payment-method-name@name	M	Req, Resp	Alphanumeric	15	This is the name of the payment method that that is chosen from the end-consumer. Currently only carrier-billing is supported.
consumer-id	O	Req, Resp	Alphanumeric	50	An id of the end-consumer in the merchant's application. E.g. account name, gamer alias, login username.

8.8.3 XML- Samples for Carrier Billing

Sample of a successful Carrier Billing debit transaction request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="EUR">10.01</requested-amount>
  <account-holder>
        <phone>00491799999999</phone>
        <address>
          <country>DE</country>
        </address>
  </account-holder>
  <order-detail>Membership</order-detail>
  <descriptor>Game online</descriptor>
  <notifications>
        <notification url="https://shop.com/notifications"/>
  </notifications>
  <payment-methods><payment-method name="carrier-billing"/></payment-methods>
  <consumer-id>myaccountname</consumer-id>
  <processing-redirect-url>http://myshop.com/processing.html</processing-redirect-url>
  <cancel-redirect-url>http://myshop.com/cancel.html</cancel-redirect-url>
  <fail-redirect-url>http://myshop.com/fail.html</fail-redirect-url>
  <success-redirect-url>http://myshop.com/success.html</success-redirect-url>
  <locale>de_DE</locale>
</payment>
```

Sample of a successful Carrier Billing debit transaction response:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <transaction-id>2109370c-beea-11e2-a615-005056a9092e</transaction-id>
  <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2014-09-05T12:05:57.415Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created."
```

```
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10.01</requested-amount>
  <account-holder>
        <phone>00491799999999</phone>
        <address>
           <country>DE</country>
        </address>
  </account-holder>
  <order-detail>Membership</order-detail>
  <descriptor>Game online</descriptor>
  <notifications>
        <notification url="https://shop.com/notifications"/>
  </notifications>
  <payment-methods>
    <payment-method url="https://securepay.mopay.com/onetime/button/start.action?..."
name="carrier-billing"/>
  </payment-methods>
  <consumer-id>myaccountname</consumer-id>
  <processing-redirect-url>http://myshop.com/processing.html</processing-redirect-url>
  <cancel-redirect-url>http://myshop.com/cancel.html</cancel-redirect-url>
  <fail-redirect-url>http://myshop.com/fail.html</fail-redirect-url>
  <success-redirect-url>http://myshop.com/success.html</success-redirect-url>
  <locale>de_DE</locale>
  <country>DE</country>
</payment>
```

Sample of a notification for a successful Carrier Billing transaction:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>048993d0-a110-11e2-9e96-0800200c9a66</merchant-account-id>
  <transaction-id>a561a25a-beea-11e2-a615-005056a9092e</transaction-id>
  <request-id>1d613bdb-8feb-4334-bb2e-ae62cdffadf1</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2014-09-05T12:09:38.452Z</completion-time-stamp>
  <statuses>
    <status code="200.0000" description="mopay:The request completed successfully."
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10.01</requested-amount>
  <account-holder>
        <phone>00491799999999</phone>
        <address>
           <country>DE</country>
        </address>
  </account-holder>
  <order-detail>Membership</order-detail>
  <descriptor>Game online</descriptor>
  <payment-methods>
    <payment-method name="carrier-billing"/>
  </payment-methods>
  <consumer-id>myaccountname</consumer-id>
  <processing-redirect-url>http://myshop.com/processing.html</processing-redirect-url>
  <cancel-redirect-url>http://myshop.com/cancel.html</cancel-redirect-url>
  <fail-redirect-url>http://myshop.com/fail.html</fail-redirect-url>
  <success-redirect-url>http://myshop.com/success.html</success-redirect-url>
  <locale>de_DE</locale>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>SzqyN8C6jtx0nxuT6t4VDfQivW8=</DigestValue>
      </Reference>
    </SignedInfo>

<SignatureValue>QcJs5pU1scWzBe7QBwY636+sdezj7ZUttkLOQxxcD4NmnUcVtFtDnZc73vjWSTL6pBsfI9v
XMGhq
qoeAZaoGpTLbEKQ5rm4Z7oKfWnijPiO4rQ4Cj4uUIi8EqCBvqCOE6NRFOkFNVWBxYOLx0wEphVxf
+xTQ3oJiw/8tbmbg7U9fUVKTWlmyJ2jMOf+1gaKxWCBTS7pUTX5qAK/o7rRgIO5DBFOhqL2eqyOk
B8V0rsYUBLflSJdeOlh4iQ1DTNhXXr5VVNEdI1Fknda+APut6gpQ3rWM+UOYypXMlMaLj2UPMTv3
PjeL7C+dNVJ+15qRfc0QL+9l/oVL5iZfO3tFKw==</SignatureValue>
    <KeyInfo>
      <X509Data>
        <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>

<X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJD
QTELMAkGA1UE
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVu
dHMxEzARBgNVBAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgN
VBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAsTCk9w
ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSaIF3JKBxf7gDAik5d8RT5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIlAMGzW7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWbIzyQuyUyDeUb
QmhVQKL0WlMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXfOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
</payment>
```

Sample of a successful Carrier Billing refund-debit transaction request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>27189fdb-ffbe-4025-ad4f-6c271c853204</merchant-account-id>
  <request-id>fc63d62c-b4d8-4c0a-9330-5a64d5c91284</request-id>
  <transaction-type>refund-debit</transaction-type>
  <parent-transaction-id>e93f28f2-7bd2-11e4-8190-00163e000157</parent-transaction-id>
  <descriptor>refund success</descriptor>
  <payment-methods>
         <payment-method name="carrier-billing"/>
  </payment-methods>
</payment>
```

Sample of a successful Carrier Billing refund transaction response:

```
<payment
       self="https://api-test.wirecard.com/engine/rest/merchants/27189fdb-ffbe-
4025-ad4f-6c271c853204/payments/ea3e72f8-7bd2-11e4-8190-00163e000157"
       xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id ref="https://api-
test.wirecard.com/engine/rest/merchants/27189fdb-ffbe-4025-ad4f-6c271c853204">
             27189fdb-ffbe-4025-ad4f-6c271c853204
   </merchant-account-id>
   <transaction-id>ea3e72f8-7bd2-11e4-8190-00163e000157</transaction-id>
   <request-id>fc63d62c-b4d8-4c0a-9330-5a64d5c91284</request-id>
   <transaction-type>refund-debit</transaction-type>
   <transaction-state>success</transaction-state>
   <completion-time-stamp>2014-12-04T16:30:56.000Z</completion-time-stamp>
   <statuses>
      <status code="201.0000" description="carrier-billing:The resource was
successfully created." severity="information"/>
   </statuses>
   <requested-amount currency="EUR">0.29</requested-amount>
   <descriptor>refund success</descriptor>
   <notifications>
      <notification url="http://shop.com/notification"/>
   </notifications>
   <payment-methods>
      <payment-method name="carrier-billing"/>
   </payment-methods>
   <api-id>elastic-api</api-id>
   <processing-redirect-url>https://api-
test.wirecard.com/shop/processing.html</processing-redirect-url>
   <cancel-redirect-url>https://api-test.wirecard.com/shop/cancel.html</cancel-
redirect-url>
   <fail-redirect-url>https://api-test.wirecard.com/shop/error.html</fail-
redirect-url>
   <success-redirect-url>https://api-test.wirecard.com/shop/success.html</success-
redirect-url>
</payment>
```

9 Direct Debit and Credit Transfer

Merchants can send payments to the Wirecard Payments Platform using various payment methods. In order to be able to create a payment using a specific payment method, the merchant has to have a valid subscription for this payment method and needs to specify it correctly inside the payment request.

9.1 Create a Payment for Direct Debit and Credit Transfer

URL	https://hostname/engine/rest/paymentmethods
Request Formats	XML
Response Formats	XML
Request Methods	POST

Direct Debit and Credit Transfer payments are a one-step process. The merchant sends in an XML request with the consumer's banking data and Wirecard stores this information and forwards it on to the bank for further processing.

9.2 SEPA

The Single Euro Payments Area (SEPA) is an initiative of the European banking industry that will make all electronic payments across the euro area as easy as domestic payments within one country. This currently includes credit transfer and direct debit. The payments are processed under the same basic conditions, rights, and obligations regardless of their location. The geographical scope of SEPA encompasses the 28 EU Member States, Iceland, Liechtenstein, Monaco, Norway, and Switzerland. The SEPA project is strongly supported by the European Commission and the European Central Bank. It is based on ISO 20022 standards.

To be able to test SEPA Direct Debit and SEPA Credit Transfer transactions, please use the following test merchant account information:

URL	https://api-test.wirecard.com/engine/rest/paymentmethods/
TEST Merchant Account ID	4c901196-eff7-411e-82a3-5ef6b6860d64
TEST Merchant Account Name	WD SEPA Test
Username	70000-APITEST-AP
Password	qD2wzQ_hrc!8

9.2.1 SEPA Direct Debit

Please see also Appendix A: Payment Methods.

The transaction type within the request must be *DEBIT or PENDING-DEBIT.*

If the request is successful, a response will be returned with status Success. This response means that the transaction has entered the Wirecard system successfully. The transaction is in a pending status waiting to be sent to the merchant's bank.

If the request is not successful, a response will be returned with status Failed. **The response always contains a status code and a description.**
*Please read this description carefully as it will help understand why the transaction request has failed and what needs to be fixed in order to send a successful transaction request.

Once the transaction has been successfully sent to and processed by the bank, a new transaction with type *DEBIT* will be created in status Success. This process usually takes up to 2 business days. The transaction will then be forwarded the Central Bank for processing.

If the transaction has been rejected by the merchant's bank, a *DEBIT* on status Failed will be created and the transaction will not be sent to the Central Bank.
The merchant will receive a notification of the *DEBIT*.

PLEASE NOTE: Although a transaction has been successfully processed by the merchant's bank, the transaction may still be reversed by the consumer's bank for reasons such as insufficient funds, account owner deceased, bank account closed, etc. In this case, a debit-return transaction will be created and matched to the original debit transaction to ensure the merchant has a complete overview for his debtor management. For a full list of debit return reasons, please refer to Appendix J: List of SEPA Direct Debit and Credit Transfer Return Codes.

The merchant can request the status of the transaction at any time by sending a “Retrieve Transaction by Transaction ID” (see Chapter 0) or “Retrieve Transaction by Request ID” (see Chapter 5.4).

A reconciliation file is available to be downloaded for merchants. This allows merchant to have the most up-to-date status for their transactions. For more information about the reconciliation file, please contact Wirecard Customer Support.

9.2.1.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method SEPA Direct Debit. If an element is marked as mandatory, it is stated whether it is mandatory in the request (Req) or the Response (Resp).

Term	Man/Opt Req/Resp	Format	Max Size	Description
account-holder.address.city	O	Alphanumeric	32	This is the end-consumer’s city
account-holder.address.country	O	Alphanumeric	3	This is the end-consumer’s country
account-holder.address.postal-code	O	Alphanumeric	16	This is the end-consumer’s postal code
account-holder.address.state	O	Alphanumeric	32	This is the end-consumer’s state
account-holder.address.street1	O	Alphanumeric	70	This is the first part of the end-consumer’s street
account-holder.address.street2	O	Alphanumeric	128	This is the second part of the end-consumer’s street.
account-holder.date-of-birth	O	Date	0	This is the end-consumer’s birthdate
account-holder.email	O	Alphanumeric	64	This is the end-consumer’s Email-Address.
account-holder.first-name	M (Req)	Alphanumeric	32	This is the first name of the end-consumer.
account-holder.gender	O	Alphanumeric	1	This is the end-consumer’s gender
account-holder.last-name	M (Req)	Alphanumeric	32	This is the last name of the end-consumer.
account-holder.phone	O	Alphanumeric	32	This is the phone number of the end-consumer
bank-account.bic	M (Req)	Alphanumeric	8 or 11	This is the Business Identifier Code of the bank of the end-consumer. Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})

bank-account.iban	M (Req)	Alphanumeric	34	This is the International Bank Account Number of the end-consumer. Allowed characters: [a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}
creditor-id	M (Req)	Alphanumeric	1,35	Identifies and allows a merchant to process SEPA Direct Debit transactions. Allowed characters: [a-zA-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{3,3}[a-zA-Z0-9]{1,28}
custom-field.field-name	O	Alphanumeric	36	This is the name for the custom field
custom-field.field-value	O	Alphanumeric	256	This is the content of the custom field. In this field the merchant can send additional information
descriptor	O	Alphanumeric	100	Description on the settlement of the account holder’s account about a transaction. For SEPA Direct Debit transactions, it will be combined with the Provider Transaction Reference ID and the merchant’s static descriptor and will appear on the consumer’s bank account statement.

<table>
<tr><td>due-date</td><td>O</td><td>Date</td><td>0</td><td>The date on which the funds will be taken from the consumer’s account for a SEPA Direct Debit transaction. This can either be sent in by the merchant or it will be calculated by Wirecard. Please refer to section <i>Due Date</i> below.</td></tr>
<tr><td>ip-address</td><td>O</td><td>Alphanumeric</td><td>15</td><td>The global (internet) IP address of the consumer’s computer.</td></tr>
<tr><td>mandate.mandate-id</td><td>M (Req)</td><td>Alphanumeric</td><td>35</td><td>The ID of the signed mandate between the merchant and the consumer. Allowed characters:[A-Za-z0-9]|[\+|\?|/|\-|:|\(|\)|\.|,|']){1,35}</td></tr>
<tr><td>mandate.signature-date</td><td>M (Req)</td><td>Date</td><td>-</td><td>The date on which the mandate was signed</td></tr>
<tr><td>merchant-account-id</td><td>M (Req)</td><td>Alphanumeric</td><td>36</td><td>Unique identifier for a merchant account</td></tr>
<tr><td>order-detail</td><td>O</td><td>Alphanumeric</td><td>65535</td><td>This is a field for details of an order filled by the merchant</td></tr>
<tr><td>order-number</td><td>O</td><td>Alphanumeric</td><td>64</td><td>This is the order number of the merchant</td></tr>
<tr><td>parent-transaction-id</td><td>O</td><td>Alphanumeric</td><td>36</td><td>Transaction ID of the first transaction of a payment</td></tr>
<tr><td>provider-transaction-reference-id</td><td>M (Resp)</td><td>Alphanumeric</td><td>10</td><td>This ID represents the end-to-end flow of a transaction and is used to identify transactions within the banking system. It is generated by Wirecard.</td></tr>
</table>

requested-amount	M (Req)	Numeric	11,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency. Min amount €0.01. Max amount €999999999.99.
requested-currency	M (Req)	Alpha	3	This is the currency of the requested-amount. Only EUR (euro) is accepted.
request-id	M (Req)	Alphanumeric	150	This is the identification number of the request on the merchant's side. **It must be unique for each request.**
status.code	M (Resp)	Alphanumeric	12	This is the code of the status of a transaction
status.description	M (Resp)	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M (Resp)	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M (Resp)	Alphanumeric	12	This is the status of a transaction
transaction-id	M (Resp)	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
transaction-type	M (Req)	Alphanumeric	30	This is the type for a transaction. For SEPA Direct Debit, only *DEBIT* is allowed.

SEPA Direct Debit Specific Fields:

The following elements should be kept in mind when sending in a SEPA Direct Debit payment request:

- Creditor Identifier
- Mandate ID
- Mandate Signature Date
- Provider Transaction Reference ID
- Due Date

Creditor Identifier

In order to be able to offer *SEPA Direct Debit* as a payment method, a merchant must apply for a Creditor Identifier. The Creditor Identifier identifies a merchant and allows consumers to be able to manage their mandates with merchants more easily.

The Creditor Identifier is only required for SEPA Direct Debit and not for SEPA Credit Transfer transactions.

A German merchant may apply for a Creditor Identifier at the Bundesbank Deutschland (http://www.bundesbank.de/Navigation/EN/Tasks/Payment_systems/SEPA/Creditor_Identifier/creditor_identifier.html?docId=25848&notFirst=true)

The following characters are allowed: [a-zA-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{3,3}[a-zA-Z0-9]{1,28}.

Mandate ID

A two-party mandate between the merchant and the debtor is required for *SEPA Direct Debit* payments. The mandate is an agreement giving the merchant permission to debit the consumer's account for the sum that the two parties agreed upon. Every mandate has an ID provided by the merchant that, when combined with the Creditor Identifier, creates a unique mandate id. The mandate reference must be sent to the Wirecard Payments Platform within the request.

The Mandate ID is only required for SEPA Direct Debit and not for SEPA Credit Transfer transactions.

PLEASE NOTE: The mandate ID may NOT be generated by the Wirecard Payments Platform. It is solely the responsibility of the merchant.

Mandate Signature Date

This is the date on which the above-mentioned mandate was signed by the consumer.

The Mandate Signature Date is only required for SEPA Direct Debit and not for SEPA Credit Transfer transactions.

Provider Transaction Reference ID

This ID provides a reference for the complete end-to-end lifecycle of a SEPA Direct Debit transaction. It is used as a reference within the banking system to ensure all transactions referencing each other (eg: a direct debit and a debit return) are matched and that the complete lifecycle of a payment is identifiable. Wirecard generates this ID for the merchant.

Due Date

The due date is the day when the funds will be cleared on the debtors bank account. Wirecard can calculate the best (earliest) due date for the Merchant. When the Due Date field in the request is left empty, Wirecard will calculate the due date for the merchant automatically.

The merchant may, however, send in a due date inside the request if a specific date is requested. This date may only be used if it passes validation. If validation is not passed, Wirecard will return a failed transaction and the transaction must be re-submitted.

PLEASE NOTE: Wirecard cannot replace an incorrect due date sent in by the merchant with a correct due date calculated by the system as this would change the content of the transaction sent in by the merchant.

Calculating a Due Date

In order to keep processing times transparent for all transactions within SEPA, a transaction's due date has been clearly defined. One-off and initial direct debit payments must be submitted by Wirecard to the paying agent five (5) banking days before the due date. Submission of recurring payments must occur two (2) banking days before the due date.

If the merchant chooses to specify a due date, it is important to ensure all payments are submitted on time and are processed correctly. Wirecard requires merchants to submit.

- one-off and first direct debit payments at least seven (7) banking days before the due date
- recurring and final direct debit payments at least four (4) banking days before the due date.

Exception for Germany

An agreement has been reached between the European Payments Council and German banks to allow a shorter due date timeframe.

For national transactions within Germany (ie: creditor's bank account AND debtor's bank account are both German) Wirecard allows:

- All debit payments (one-off, first, recurring, and final) to be submitted three (3) banking days before the due date.

PLEASE NOTE: This is not a requirement. It is an option. If a merchant chooses to send in a transaction later than 3 days before the due date, this is possible. Wirecard will always validate a due date if one is sent in with the transaction request.

The transaction request must be received by the Wirecard system no earlier than 14 calendar days before the due date.

It is also necessary for the merchant to take into consideration the European Central Bank's Target 2 Calendar which specifies all non-banking days. These include:

- Saturdays and Sundays
- New Year's Day
- Good Friday
- Easter Monday
- 1 May (Labour Day)
- Christmas Day
- 26 December

The Due Date is only required for SEPA Direct Debit and not for SEPA Credit Transfer transactions.

For more information about recurring SEPA Direct Debit payments, please refer to section 7.2.1.2 XML - Samples for SEPA Direct Debit

9.2.1.2 XML - Samples for SEPA Direct Debit

Successful Pending-Debit Request

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
                    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
account-id>
            <request-id>7f55aacb-3e15-4185-b80f-1e0ad5b51d6c</request-id>
   <transaction-type>pending-debit</transaction-type>
   <requested-amount currency="EUR">10.01</requested-amount>
            <account-holder>
                        <first-name>John</first-name>
                        <last-name>Doe</last-name>
            </account-holder>
    <payment-methods>
                        <payment-method name="sepadirectdebit"/>
    </payment-methods>
    <bank-account>
                        <iban>DE42512308000000060004</iban>
                        <bic>WIREDEMMXXX</bic>
    </bank-account>
    <mandate>
                        <mandate-id>12345678</mandate-id>
                        <signed-date>2013-09-24</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ09999999999</creditor-id>
</payment>
```

Successful Pending-Debit Response (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>3971c2d8-250f-11e3-8d4b-005056a97162</transaction-id>
  <request-id>7f55aacb-3e15-4185-b80f-1e0ad5b51d6c</request-id>
  <transaction-type>pending-debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2013-09-24T11:48:27.528Z</completion-time-stamp>
  <statuses>
     <status code="201.0000" description="The resource was successfully created."
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10.01</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit"/>
  </payment-methods>
  <bank-account>
    <iban>DE42512308000000060004</iban>
    <bic>WIREDEMMXXX</bic>
  </bank-account>
  <mandate>
    <mandate-id>12345678</mandate-id>
    <signed-date>2013-09-24</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <due-date>2013-10-03</due-date>
  <provider-transaction-reference-id>33F7A4D125</provider-transaction-reference-id>
```

```
        </payment>
```

Failed Pending-Debit Request (account holder section missing)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
                <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-
        id>
                <request-id>6bc1f033-ed85-4363-955c-94d70b55f57f</request-id>
                <transaction-type>pending-debit</transaction-type>
                <requested-amount currency="EUR">10.01</requested-amount>
                <payment-methods>
                            <payment-method name="sepadirectdebit"/>
                </payment-methods>
                <bank-account>
                            <iban>DE42512308000000060004</iban>
                            <bic>WIREDEMMXXX</bic>
                </bank-account>
                <mandate>
                            <mandate-id>12345678</mandate-id>
                            <signed-date>2013-09-24</signed-date>
                </mandate>
                <creditor-id>DE98ZZZ09999999999</creditor-id>
        </payment>
```

Failed Pending-Debit Response (for the previous request)

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
          <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
          <transaction-id>4185215e-250f-11e3-8d4b-005056a97162</transaction-id>
          <request-id>6bc1f033-ed85-4363-955c-94d70b55f57f</request-id>
          <transaction-type>pending-debit</transaction-type>
          <transaction-state>failed</transaction-state>
          <completion-time-stamp>2013-09-24T11:48:40.583Z</completion-time-stamp>
          <statuses>
            <status code="400.1007" description="The account holder information has not
        been provided.  Please check your input and try again.  " severity="error"/>
          </statuses>
          <requested-amount currency="EUR">10.01</requested-amount>
          <payment-methods>
            <payment-method name="sepadirectdebit"/>
          </payment-methods>
          <bank-account>
            <iban>DE42512308000000060004</iban>
            <bic>WIREDEMMXXX</bic>
          </bank-account>
          <mandate>
            <mandate-id>12345678</mandate-id>
            <signed-date>2013-09-24</signed-date>
          </mandate>
          <creditor-id>DE98ZZZ09999999999</creditor-id>
        </payment>
```

7.3.1 SEPA Direct Debit Recurring Transactions

The Wirecard Payments Platform allows a merchant to submit recurring payments for SEPA Direct Debit transactions.

To submit a recurring transaction, the merchant must submit a request with the transaction type *DEBIT or PENDING-DEBIT* including the PERIODIC TYPE element with type “recurring” or “installment”. Additionally, the sub-element SEQUENCE TYPE with one of the following sequences must be submitted:

- **first**: The first transaction in a series of recurring transactions
- **recurring**: A transaction that is part of a series of recurring transactions.
- **final**: The final transaction in a series of recurring transactions. A payment with this sequence type completes a chain of recurring payments.

Periodic Type

Indicates how and why a Payment occurs more than once. 'Installment': one in a set that completes a financial transaction and 'recurring': one in a set that occurs repeatedly, such as a subscription. Both ‘recurring’ and ‘installment’ transactions will be processed the same way. The differentiation is for the merchant’s use only and which of the two periodic types is chosen depends on the merchant’s business model.

Due Date Calculation

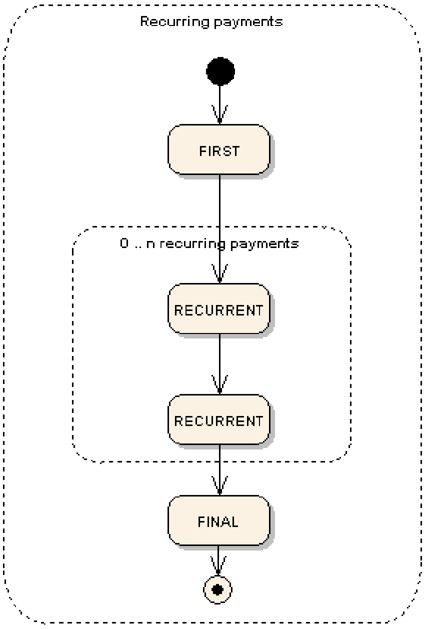
The Due Date calculation of a recurring transaction depends on the sequence type of that transaction. Please refer to section 7.2.1.1 which defines how to calculate a due date for each sequence type.

Restrictions

To ensure proper processing, please take into consideration the following restrictions for SEPA Direct Debit recurring transactions:

- “Recurring” and “final” transactions are required to be matched to the “first” transaction. Therefore, it is required that a parent-transaction ID referring to a *PENDING-DEBIT* transaction with sequence type “first” is provided in the transaction request.
- A request with sequence type “first” may only contain a parent-transaction ID referring to another transaction which has been submitted without a periodic type.
- A “recurring” or “final” transaction may only be referenced to a parent “first” *PENDING-DEBIT* if it is on status SUCCESS.
- Where a “first” pending-debit has been voided (see section 7.6.1 SEPA Void Functionality), a subsequent “recurring” or “final” transaction will not be accepted.
- After a transaction with sequence type “final” has been submitted, it is not possible to submit another “recurring” or “final” transaction for this series.
- Only one periodic type may be used for the complete series of recurring payments. This means that the periodic type in a subsequent transaction must match the periodic type sent in the “first” request.
- The Account Holder IBAN, Account Holder BIC, Account Holder Name, Creditor ID, Mandate ID, and Mandate Signature Date must be the same for “recurring” and “final” transactions as was provided in the “first” transaction.

The following figure describes possible flow of sequence types.



Referencing to a "FIRST" Recurring Transaction

A merchant must reference a pending-debit with sequence type "recurring" or "final" to an already existing pending-debit with sequence type "first" by entering the transaction ID from the "first" transaction into the parent-transaction-id in the "recurring" or "final" pending-debit request. This results in the following:

- Transactions will be referenced to each other and considered as part of the same group of transactions
- Fields provided in the "first" pending-debit will be copied into the "recurring" or "final" pending-debit

If the merchant does not want any information to be copied, the information must be completely re-submitted as required for a regular pending-debit request.

The merchant may specify a new value for a field in a "recurring" or "final" pending-debit. However, the rest of the fields will still be copied from the "first" pending-debit.

NOTE: There are exceptions to the point above.

- If the merchant specifies a new value within any of the following sections for a "recurring" or "final" pending-debit, the information previously submitted for any of the fields within this section in the "first" pending-debit will NOT be copied:
    - bankaccount
    - paymentmethods
    - shipping
    - mandate
    - airline industry
    - cruise industry

e.g. All Fields Copied:
If Account Holder First Name, Account Holder Last Name, and Account Holder Gender are provided in the "first" transaction and a new Account Holder First Name is sent in the "recurring" or "final" transaction without an Account Holder Last Name and Gender, these fields will be copied from the "first" transaction

First Pending-Debit Request:

```
    <account-holder>
            <first-name>John</first-name>
```

```
        <last-name>Doe</last-name>
        <gender>M</gender>
</account-holder>
```

First Pending-Debit Response: SUCCESS

```
<account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <gender>M</gender>
</account-holder>
```

Recurring or Final Pending-Debit Request:

```
<account-holder>
        <first-name>James</first-name>
        <last-name></last-name>
        <gender></gender>
</account-holder>
```

Pending-Debit Response: SUCCESS

```
<account-holder>
        <first-name>James</first-name>
        <last-name>Doe</last-name>
        <gender>M</gender>
</account-holder>
```

e.g. All Fields NOT Copied:

First Pending-Debit Request:

```
<mandate>
        <mandate-id>12345678</mandate-id>
        <signed-date>2013-09-24</signed-date>
</mandate>
```

First Pending-Debit Response: SUCCESS

```
<mandate>
        <mandate-id>12345678</mandate-id>
        <signed-date>2013-09-24</signed-date>
</mandate>
```

Recurring or Final Pending-Debit Request:

```
<mandate>
        <mandate-id>987654321</mandate-id>
        <signed-date></signed-date>
</mandate>
```

Pending-Debit Response: **FAILED** Missing Mandate Signed Date

```
<mandate>
        <mandate-id>987654321</mandate-id>
        <signed-date></signed-date>
</mandate>
```

Because new information, a mandate ID, has been provided in the section “mandate” in the pending-debit request, the signed-date is not copied. The pending-debit fails as it is missing a signed-date.

7.3.1.1 Fields

In addition to a one-off SEPA Direct Debit Transaction (see section Fields), the following fields are required for a recurring SEPA Direct Debit transaction.

Term	Man/Opt Req/Resp	Format	Max Size	Description
parent-transaction-id	M* (req) *mandatory for “recurring” and “final” txs	Alphanumeric	36	Transaction ID of the first transaction of a payment
periodic.periodic-type	M (req)	Alpha	9	Indicates how and why a Payment occurs more than once. Only two possible values: 'recurring' or ‘installment’
periodic.periodic-type.sequence-type	M (req)	Alpha	9	Indicates the sequence of the recurring transaction. Possible values: ‘first’, ‘recurring’ or ‘final’

7.3.1.2 XML - Samples for Recurring SEPA Direct Debit

Successful Pending-Debit Request “First”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">

    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
    <request-id>67366dbf-c68c-4f4e-a14b-69db83fbdd20</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="EUR">20.02</requested-amount>
    <account-holder>
                    <first-name>John</first-name>
                    <last-name>Doe</last-name>
    </account-holder>
    <payment-methods>
                    <payment-method name="sepadirectdebit"/>
    </payment-methods>
    <bank-account>
                    <iban>DE42512308000000060004</iban>
                    <bic>WIREDEMMXXX</bic>
    </bank-account>
    <mandate>
                    <mandate-id>12345678</mandate-id>
                    <signed-date>2013-12-19</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ09999999999</creditor-id>
    <periodic>
            <periodic-type>recurring</periodic-type>
            <sequence-type>first</sequence-type>
    </periodic>
</payment>
```

Successful Pending-Debit Response “First”(for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>e6604f91-663c-11e3-a07b-18037336c0b3</transaction-id>
  <request-id>67366dbf-c68c-4f4e-a14b-69db83fbdd20</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2013-12-19T10:29:02.000Z</completion-time-stamp>
  <statuses>
      <status code="201.0000" description="The resource was successfully created."
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">20.02</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit"/>
  </payment-methods>
  <bank-account>
    <iban>DE42512308000000060004</iban>
    <bic>WIREDEMMXXX</bic>
  </bank-account>
  <mandate>
    <mandate-id>12345678</mandate-id>
    <signed-date>2013-12-19</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <due-date>2014-01-02</due-date>
  <periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>first</sequence-type>
  </periodic>
  <provider-transaction-reference-id>5A00C85484</provider-transaction-reference-id>
</payment>
```

Successful Pending-Debit Request “Recurring”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">

   <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
   <request-id>55566dbf-c68c-4f4e-a14b-69db83fbd555</request-id>
   <transaction-type>debit</transaction-type>
   <parent-transaction-id> e6604f91-663c-11e3-a07b-18037336c0b3</parent-transaction-id>
   <payment-methods>
           <payment-method name="sepadirectdebit"/>
   </payment-methods>
   <periodic>
          <periodic-type>recurring</periodic-type>
          <sequence-type>recurring</sequence-type>
   </periodic>
</payment>
```

Successful Pending-Debit Response “Recurring” (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>e6604f91-663c-11e3-a07b-18037336c0b3</transaction-id>
  <request-id>55566dbf-c68c-4f4e-a14b-69db83fbd555</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2013-12-19T10:29:02.000Z</completion-time-stamp>
  <statuses>
     <status code="201.0000" description="The resource was successfully created."
severity="information"/>
  </statuses>
  <requested-amount currency="EUR">20.02</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit"/>
  </payment-methods>
  <bank-account>
    <iban>DE42512308000000060004</iban>
    <bic>WIREDEMMXXX</bic>
  </bank-account>
  <mandate>
    <mandate-id>12345678</mandate-id>
    <signed-date>2013-12-19</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <due-date>2014-01-02</due-date>
  <periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>recurring</sequence-type>
  </periodic>
  <provider-transaction-reference-id>6A11C85484</provider-transaction-reference-id>
</payment>
```

Failed Pending-Debit Request with Recurring (missing parent transaction ID)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">

   <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
   <request-id>05913099-b20e-4e2b-a8b0-e28fa9ea700c</request-id>
   <transaction-type>debit</transaction-type>
   <requested-amount currency="EUR">20.02</requested-amount>
   <account-holder>
                   <first-name>John</first-name>
                   <last-name>Doe</last-name>
   </account-holder>
   <payment-methods>
                   <payment-method name="sepadirectdebit"/>
   </payment-methods>
   <bank-account>
                   <iban></iban>
                   <bic>WIREDEMMXXX</bic>
   </bank-account>
   <mandate>
```

```
                    <mandate-id>12345678</mandate-id>
                    <signed-date>2013-12-19</signed-date>
      </mandate>
      <creditor-id>DE98ZZZ09999999999</creditor-id>
      <periodic>
            <periodic-type>recurring</periodic-type>
            <sequence-type>recurring</sequence-type>
      </periodic>

    </payment>
```

Failed Pending-Debit Response with Recurring (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>5d21ee54-663d-11e3-a07b-18037336c0b3</transaction-id>
  <request-id>05913099-b20e-4e2b-a8b0-e28fa9ea700c</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>failed</transaction-state>
  <completion-time-stamp>2013-12-19T10:32:21.000Z</completion-time-stamp>
  <statuses>
    <status code="400.1021" description=" No Parent Transaction Id.The Parent
Transaction Id is required, and not provided." severity="error"/>
  </statuses>
  <requested-amount currency="EUR">20.02</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit"/>
  </payment-methods>
  <bank-account>
    <iban/>
    <bic>WIREDEMMXXX</bic>
  </bank-account>
  <mandate>
    <mandate-id>12345678</mandate-id>
    <signed-date>2013-12-19</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>recurring</sequence-type>
  </periodic>
</payment>
```

7.4.1 SEPA Direct Debit Authorization Transactions

The Wirecard Payments Platform allows a merchant to submit authorization transactions for SEPA Direct Debit payments.

To submit an authorization transaction, the merchant must submit a request with the transaction type *AUTHORIZATION* and the payment-method name *SEPADIRECTDEBIT*.

The authorization transaction type is not mandatory to use for SEPA Direct Debit transactions. A transaction with type authorization is not sent on for further processing to the provider. It remains in the Wirecard Payments Platform for future reference.

If the merchant sends a transaction request with transaction type authorization, the merchant will receive a Transaction ID in the response which can be used to reference future pending-debit transactions.

A few reasons a merchant may decide to use the authorization transaction are, among others, the following:

- to split a payment into two or more debits referencing one original authorization; or
- to validate an end-customer's banking details before sending in a pending-debit at a different time

Referencing to an Authorization

A merchant is able to reference a pending-debit to an authorization by entering the authorization's transaction ID into the parent-transaction-id in the pending-debit request. This results in the following:

- Transactions will be referenced to each other and considered as part of the same Payment
- Fields provided in the authorization will be copied into the pending-debit

If the merchant does not want any information to be copied, the information must be completely re-submitted as required for a regular pending-debit request.

The merchant may specify a new value for a field in a pending-debit. However, the rest of the fields will still be copied from the authorization.

NOTE: There are exceptions to the point above.

- If the merchant specifies a new value within any of the following sections for a "recurring" or "final" pending-debit, the information previously submitted for any of the fields within this section in the authorization will NOT be copied:
  - bankaccount
  - paymentmethods
  - shipping
  - mandate
  - airline industry
  - cruise industry

Authorization Request:

```
<bank-account>
        <iban>DE42512308000000060004</iban>
        <bic>WIREDEMMXXX</bic>
</bank-account>
```

Authorization Response: SUCCESS

```
<bank-account>
        <iban>DE42512308000000060004</iban>
        <bic>WIREDEMMXXX</bic>
</bank-account>
```

Pending-Debit Request:

```
<bank-account>
        <iban>DE20512308000000060005</iban>
        <bic></bic>
</bank-account>
```

Pending-Debit Response: **FAILED** – Missing BIC

```
<bank-account>
        <iban> DE20512308000000060005</iban>
        <bic></bic>
</bank-account>
```

Because new information, a new bank account, has been provided in the section "bank-account" in the pending-debit request, the BIC is not copied. The pending-debit fails as it is missing a BIC.

7.4.1.1 Fields

The following elements are mandatory for sending a request/response for an authorization transaction for the payment method SEPA Direct Debit.

NOTE: Some of the elements that are mandatory for a SEPA direct debit transaction are not mandatory for a SEPA authorization transaction.
For example, IBAN and BIC are optional. If a merchant requires an IBAN and BIC to be validated, they must be sent in the request. If they are sent within the request the system validates their formal correctness.

Term	Man/Opt Req/Resp	Format	Max Size	Description
account-holder.first-name	M (Req)	Alphanumeric	32	This is the first name of the end-consumer.
account-holder.last-name	M (Req)	Alphanumeric	32	This is the last name of the end-consumer.
bank-account.bic	O (Req)	Alphanumeric	8 or 11	This is the Business Identifier Code of the bank of the end-consumer. If it is sent within the request if will be validated to the formal correctness. Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})?)
bank-account.iban	O (Req)	Alphanumeric	34	This is the International Bank Account Number of the end-consumer. If it is sent within the request if will be validated to the formal correctness. Allowed characters: [a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}
merchant-account-id	M (Req)	Alphanumeric	36	Unique identifier for a merchant account

provider-transaction-reference-id	M (Resp)	Alphanumeric	10	This ID represents the end-to-end flow of a transaction and is used to identify transactions within the banking system. It is generated by Wirecard.
requested-amount	M (Req)	Numeric	11,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency. Min amount €0.01. Max amount €999999999.99.
requested-currency	M (Req)	Alpha	3	This is the currency of the requested-amount. Only EUR (euro) is accepted.
request-id	M (Req)	Alphanumeric	150	This is the identification number of the request on the merchant's side. **It must be unique for each request.**
status.code	M (Resp)	Alphanumeric	12	This is the code of the status of a transaction
status.description	M (Resp)	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M (Resp)	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M (Resp)	Alphanumeric	12	This is the status of a transaction
transaction-id	M (Resp)	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
transaction-type	M (Req)	Alphanumeric	30	This is the type for a transaction. For a SEPA Authorization, only *AUTHORIZATION* is allowed.

7.4.1.2 XML – Sample for AUTHORIZATION SEPA Direct Debit

Successful Authorization Request

```
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <payment xmlns="http://www.elastic-payments.com/schema/payment">
     <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
     <request-id>7f55aacb-3e15-4185-b80f-1e0ad5b51d6c</request-id>
     <transaction-type>authorization</transaction-type>
     <requested-amount currency="EUR">10.01</requested-amount>
     <account-holder>
             <first-name>John</first-name>
             <last-name>Doe</last-name>
     </account-holder>
     <payment-methods>
             <payment-method name="sepadirectdebit"/>
     </payment-methods>
     <bank-account>
             <iban>DE42512308000000060004</iban>
             <bic>WIREDEMMXXX</bic>
     </bank-account>
     </payment>
```

Successful Authorization Response (for the previous request)

```
   <payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id> eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
    <transaction-id>3971c2d8-250f-11e3-8d4b-005056a97162</transaction-id>
    <request-id>7f55aacb-3e15-4185-b80f-1e0ad5b51d6c</request-id>
    <transaction-type>authorization</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2013-09-24T11:48:27.528Z</completion-time-stamp>
    <statuses>
    <status code="201.0000" description="The resource was successfully created."
     severity="information"/>
     </statuses>
     <requested-amount currency="EUR">10.01</requested-amount>
     <account-holder>
     <first-name>John</first-name>
     <last-name>Doe</last-name>
     </account-holder>
     <payment-methods>
    <payment-method name="sepadirectdebit"/>
    </payment-methods>
    <bank-account>
    <iban>DE42512308000000060004</iban>
    <bic>WIREDEMMXXX</bic>
    <provider-transaction-reference-id>33F7A4D125</provider-transaction-reference-id>
  </payment>
```

Failed Authorization Request (A Provider Transaction Reference ID is provided. This is not allowed)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
   <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
   <request-id>d0076771-ff3c-4494-8687-22eecea701a8</request-id>
```

```
    <transaction-type>authorization</transaction-type>
    <requested-amount currency="EUR">15.55</requested-amount>
         <account-holder>
                 <first-name>John</first-name>
                 <last-name>Doe</last-name>
         </account-holder>
    <payment-methods>
                 <payment-method name="sepadirectdebit"/>
    </payment-methods>
    <bank-account>
                 <iban>DE42512308000000060004</iban>
                 <bic>WIREDEMMXXX</bic>
    </bank-account>
    <provider-transaction-reference-id>68E34C9581</provider-transaction-reference-id>
</payment>
```

Failed Authorization Response (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>4871ba42-b4cd-11e3-b68c-18037336c0b3</transaction-id>
  <request-id>d0076771-ff3c-4494-8687-22eecea701a8</request-id>
  <transaction-type>authorization</transaction-type>
  <transaction-state>failed</transaction-state>
  <completion-time-stamp>2014-03-26T09:59:17.000Z</completion-time-stamp>
  <statuses>
    <status code="400.1081" description="No Bank Account IBAN.  The Bank Account IBAN
information has not been provided." severity="error"/>
  </statuses>
  <requested-amount currency="EUR">15.55</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit"/>
  </payment-methods>
  <bank-account>
    <bic>WIREDEMMXXX</bic>
  </bank-account>
  <provider-transaction-reference-id>68E34C9581</provider-transaction-reference-id>
</payment>
```

7.4.1.3 XML – Sample for SEPA Authorization and subsequent Pending Direct Debit Request

Auth Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
     <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
       <request-id>136385050190</request-id>
         <transaction-type>authorization</transaction-type>
         <requested-amount currency="EUR">15.55</requested-amount>
                <account-holder>
                               <first-name>john</first-name>
                               <last-name>dbdushdwjsjusj</last-name>
                </account-holder>
     <payment-methods>
                                <payment-method name="sepadirectdebit"/>
     </payment-methods>
      <bank-account>
                               <iban>DE42512308000000060004</iban>
                               <bic>WIREDEMMXXX</bic>
     </bank-account>
</payment>
```

Auth Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
    <transaction-id>ed219c8c-9c02-11e4-8f3a-d4bed9896920</transaction-id>
    <request-id>136385050190</request-id>
    <transaction-type>authorization</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-01-14T16:35:07.000+01:00</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully created."
severity="information" />
    </statuses>
    <requested-amount currency="EUR">15.55</requested-amount>
    <account-holder>
        <first-name>john</first-name>
        <last-name>dbdushdwjsjusj</last-name>
    </account-holder>
    <payment-methods>
        <payment-method name="sepadirectdebit" />
    </payment-methods>
    <bank-account>
        <iban>DE42512308000000060004</iban>
        <bic>WIREDEMMXXX</bic>
    </bank-account>
    <provider-transaction-reference-id>C3EF061DD6</provider-transaction-reference-id>
</payment>
```

Pending-debit Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
     <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
       <request-id>92788956800121178</request-id>
         <transaction-type>pending-debit</transaction-type>
         <parent-transaction-id>ed219c8c-9c02-11e4-8f3a-d4bed9896920</parent-transaction-id>
         <requested-amount currency="EUR">00.11</requested-amount>
                <account-holder>
                               <first-name>iikssjw</first-name>
                               <last-name>wgeffeossddhse</last-name>
                               <email>kjioj@yahoo.co.uk</email>
                               <address>
                                               <street1>cizfpeapostr 12</street1>
                                               <city>sgafahahCity</city>
                                               <country>DE</country>
                               </address>
                </account-holder>
                <!-- optional
                <order-number>4509334</order-number> -->
                <!-- optional
                <descriptor>test</descriptor> -->
     <payment-methods>
                               <payment-method name="sepadirectdebit"/>
     </payment-methods>
     <bank-account>
                <iban>HU29117080012054779400000000</iban>
                <bic>GENODEF1OGK</bic>
     </bank-account>
     <mandate>
                               <mandate-id>12345678</mandate-id>
                               <!--signed-date>2015-01-14</signed-date-->
                               <signed-date>2014-05-06</signed-date>
     </mandate>
     <creditor-id>DE98ZZZ09999999999</creditor-id>
                <!-- optional
                <cancel-redirect-url>http://sandbox-
engine.thesolution.com/shop urlViaRequest/cancel.html</cancel-redirect-url>-->
                <!--due-date>2014-05-11</due-date-->
                <!--due-date>2015-01-14</due-date-->
     <consumer>
                               <first-name></first-name>
                               <last-name></last-name>
     </consumer>
</payment>
```

Pending-debit Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
  <transaction-id>ed82f23f-9c02-11e4-8f3a-d4bed9896920</transaction-id>
  <request-id>92788956800121178</request-id>
  <transaction-type>pending-debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-01-14T16:35:08.000+01:00</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information" />
  </statuses>
  <requested-amount currency="EUR">0.11</requested-amount>
  <parent-transaction-id>ed219c8c-9c02-11e4-8f3a-d4bed9896920</parent-transaction-id>
  <account-holder>
    <first-name>iikssjw</first-name>
    <last-name>wgeffeossddhse</last-name>
    <email>kjioj@yahoo.co.uk</email>
    <address>
      <street1>cizfpeapostr 12</street1>
      <city>sgafahahCity</city>
      <country>DE</country>
    </address>
  </account-holder>
  <payment-methods>
    <payment-method name="sepadirectdebit" />
  </payment-methods>
  <bank-account>
    <iban>HU29117080012054779400000000</iban>
    <bic>GENODEF1OGK</bic>
  </bank-account>
  <mandate>
    <mandate-id>12345678</mandate-id>
    <signed-date>2014-05-06</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <due-date>2015-01-23</due-date>
  <consumer>
```

```
      <first-name />
      <last-name />
    </consumer>
    <provider-transaction-reference-id>694E9B4DA3</provider-transaction-reference-id>
  </payment>
```

7.5.1 SEPA Credit Transfer

Please see also Appendix A: Payment Methods.

The transaction type within the request must be *CREDIT or PENDING-CREDIT.*

If the request is successful, a response will be returned with status Success. This response means that the transaction has entered the Wirecard system successfully. The transaction is in a pending status waiting to be sent to the merchant’s bank.

If the request is not successful, a response will be returned with status Failed. **The response always contains a status code and a description.**
*Please read this description carefully as it will help understand why the transaction request has failed and what needs to be fixed in order to send a successful transaction request.

Once the transaction has been successfully sent to and processed by the bank, a new transaction with type *CREDIT* will be created in status Success. This process usually takes up to 2 business days. The transaction will then be forwarded the Central Bank for processing.

If the transaction has been rejected by the merchant’s bank, a *CREDIT* on status Failed will be created and the transaction will not be sent to the Central Bank.

The merchant will receive a notification of the *CREDIT*. For more information on this notification, please see section 7.6.1.2 XML Samples for SEPA Void Tr.

Recurring Transactions are not supported for SEPA Credit Transfer.

PLEASE NOTE: Although a transaction has been successfully processed by the merchant’s bank, the transaction may still be reversed by the consumer’s bank for reasons such as insufficient funds, account owner deceased, bank account closed, etc. In this case, a credit-return transaction will be created and matched to the original credit transaction to ensure the merchant has a complete overview for his debtor management. For a full list of credit return reasons, please refer to Appendix J: List of SEPA Direct Debit and Credit Transfer Return Codes.

The merchant can request the status of the transaction at any time by sending a “Retrieve Transaction by Transaction ID” (see Chapter 0) or “Retrieve Transaction by Request ID” (see Chapter 5.4).

A reconciliation file is also available to be downloaded for merchants. This allows merchant to have the most up-to-date status for their transactions. For more information about the reconciliation file, please contact Wirecard Customer Support.

Refunds:
The SEPA Credit Transfer payment method may be used as a method for merchants to refund consumers. If it is being used as a way for the merchant to refund a specific debit transaction, it is recommended that a parent-transaction ID (ie: the transaction ID of the successful DEBIT) is included in the request in order to be able to match the “refund” to the original transaction.

7.5.1.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method SEPA Credit Transfer:

Term	Man /Opt	Format	Max Size	Description
account-holder.address.city	O	Alphanumeric	32	This is the end-consumer's city
account-holder.address.country	O	Alphanumeric	3	This is the end-consumer's country
account-holder.address.postal-code	O	Alphanumeric	16	This is the end-consumer's postal code
account-holder.address.state	O	Alphanumeric	32	This is the end-consumer's state
account-holder.address.street1	O	Alphanumeric	70	This is the first part of the end-consumer's street
account-holder.address.street2	O	Alphanumeric	128	This is the second part of the end-consumer's street.
account-holder.date-of-birth	O	Date	0	This is the end-consumer's birthdate
account-holder.email	O	Alphanumeric	64	This is the end-consumer's Email-Address.
account-holder.first-name	M (Req)	Alphanumeric	32	This is the first name of the end-consumer.
account-holder.gender	O	Alphanumeric	1	This is the end-consumer's gender
account-holder.last-name	M (Req)	Alphanumeric	32	This is the last name of the end-consumer.
account-holder.phone	O	Alphanumeric	32	This is the phone number of the end-consumer
bank-account.bic	M (Req)	Alphanumeric	8 or11	This is the Business Identifier Code of the bank of the end-consumer. Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})?)

bank-account.iban	M (Req)	Alphanumeric	34	This is the International Bank Account Number of the end-consumer. Allowed characters: [a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}
custom-field.field-name	O	Alphanumeric	36	This is the name for the custom field
custom-field.field-value	O	Alphanumeric	256	This is the content of the custom field. In this field the merchant can send additional information
descriptor	O	Alphanumeric	100	Description on the settlement of the account holder’s account about a transaction. For SEPA Credit Transfer transactions, it will be combined with the Provider Transaction Reference ID and the merchant’s static descriptor and will appear on the consumer’s bank account statement.
ip-address	O	Alphanumeric	15	The global (internet) IP address of the consumer’s computer.
merchant-account-id	M (Req)	Alphanumeric	36	Unique identifier for a merchant account
order-detail	O	Alphanumeric	65535	This is a field for details of an order filled by the merchant
order-number	O	Alphanumeric	64	This is the order number of the merchant
parent-transaction-id	O	Alphanumeric	36	Transaction ID of the first transaction of a payment

provider-transaction-reference-id	M (Resp)	Alphanumeric	10	This ID represents the end-to-end flow of a transaction and is used to identify transactions within the banking system. It is generated by Wirecard.
requested-amount	M (Req)	Numeric	11,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency. Min amount €0.01. Max amount €999999999.99.
requested-currency	M (Req)	Alpha	3	This is the currency of the requested-amount. Only EUR (euro) is accepted.
request-id	M (Req)	Alphanumeric	150	This is the identification number of the request on the merchant’s side. **It must be unique for each request.**
status.code	M (Resp)	Alphanumeric	12	This is the code of the status of a transaction
status.description	M (Resp)	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M (Resp)	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M (Resp)	Alphanumeric	12	This is the status of a transaction
transaction-id	M (Resp)	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
transaction-type	M (Req)	Alphanumeric	30	This is the type for a transaction. For SEPA Credit Transfer only *CREDIT* is allowed in the initial request.

SEPA Credit Transfer Specific Fields:

The following elements should be kept in mind when sending in a SEPA Direct Debit payment request:

- Provider Transaction Reference ID
- Execution Date

Provider Transaction Reference ID

This ID provides a reference for the complete end-to-end lifecycle of a SEPA Direct Debit transaction. It is used as a reference within the banking system to ensure all transactions referencing each other (eg: a direct debit and a debit return) are matched and that the complete lifecycle of a payment is identifiable. Wirecard generates this ID for the merchant.

Execution Date

This is the date on which the transfer will be executed. According to SEPA Regulations, it is required to be one (1) business day after it has entered the banking system. This date is set by Wirecard during processing and is not part of the request or the response.

7.5.1.2 XML Samples for SEPA Credit Transfer

Successful Pending-Credit Request

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
    <request-id>760fc06c-eca1-40e0-9029-79e49afce865</request-id>
    <transaction-type>pending-credit</transaction-type>
    <requested-amount currency="EUR">10.01</requested-amount>
            <account-holder>
                        <first-name>John</first-name>
                        <last-name>Doe</last-name>
            </account-holder>
    <payment-methods>
                        <payment-method name="sepacredit"/>
    </payment-methods>
    <bank-account>
                        <iban>DE42512308000000060004</iban>
                        <bic>WIREDEMMXXX</bic>
    </bank-account>
</payment>
```

Successful Pending- Credit Response (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
                    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
account-id>
            <transaction-id>cb71aaa6-4088-11e3-95d5-1803733b8f04</transaction-id>
            <request-id>760fc06c-eca1-40e0-9029-79e49afce865</request-id>
            <transaction-type>pending-credit</transaction-type>
            <transaction-state>success</transaction-state>
            <completion-time-stamp>2013-10-29T10:56:46.000Z</completion-time-stamp>
            <statuses>
                          <status code="201.0000" description="The resource was
successfully created." severity="information"/>
            </statuses>
            <requested-amount currency="EUR">10.01</requested-amount>
            <account-holder>
                        <first-name>John</first-name>
```

```
                <last-name>Doe</last-name>
        </account-holder>
        <payment-methods>
                <payment-method name="sepacredit"/>
        </payment-methods>
        <bank-account>
                <iban>DE42512308000000060004</iban>
                <bic>WIREDEMMXXX</bic>
        </bank-account>
          <provider-transaction-reference-id>59488C8632</provider-transaction-reference-
id>
</payment>
```

Failed Pending- Credit Request (missing IBAN)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
    <request-id>5b9cc0ce-519e-4a0a-a30f-cd4f76b62c5f</request-id>
    <transaction-type>pending-credit</transaction-type>
    <requested-amount currency="EUR">10.01</requested-amount>

        <account-holder>
                <first-name>John</first-name>
                <last-name>Doe</last-name>
        </account-holder>
    <payment-methods>
                <payment-method name="sepacredit"/>
    </payment-methods>
    <bank-account>
                <bic>WIREDEMMXXX</bic>
    </bank-account>
</payment>
```

Failed Pending-Credit Response (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
              <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
account-id>
        <transaction-id>fd3ede62-4088-11e3-95d5-1803733b8f04</transaction-id>
        <request-id>5b9cc0ce-519e-4a0a-a30f-cd4f76b62c5f</request-id>
        <transaction-type>pending-credit</transaction-type>
        <transaction-state>failed</transaction-state>
        <completion-time-stamp>2013-10-29T10:58:09.000Z</completion-time-stamp>
        <statuses>
                 <status code="400.1081" description="The Bank Account IBAN
information has not been provided.  Please check your input and try again."
severity="error"/>
        </statuses>
        <requested-amount currency="EUR">10.01</requested-amount>
        <account-holder>
                <first-name>John</first-name>
                <last-name>Doe</last-name>
        </account-holder>
        <payment-methods>
                <payment-method name="sepacredit"/>
        </payment-methods>
        <bank-account>
                <bic>WIREDEMMXXX</bic>
        </bank-account>
        </payment>
```

7.6.1 SEPA Void Functionality

The Wirecard Payments Platform allows a merchant to void payments for SEPA Direct Debit and SEPA Credit transactions before they are forwarded to the bank.

To submit a void transaction for an existing SEPA pending-debit transaction, the merchant must submit a request with the transaction type *VOID-DEBIT or VOID-PENDING-DEBIT* and the payment-method name *SEPADIRECTDEBIT.*

To submit a void transaction for an existing SEPA pending-credit transaction a request with the transaction type *VOID-CREDIT* or *VOID-PENDING-CREDIT* and the payment-method name *SEPACREDIT* must be submitted.

Restrictions
To ensure proper processing, please take into consideration the following restrictions for SEPA void transactions:

- A pending-debit or pending-credit transaction may only be voided before the transaction is batched into a file and sent to the bank for processing. Transactions are sent for processing at around 10:00 UTC.
- If a void request is sent after the transaction has been sent to the bank, a failed response will be returned. Once a corresponding Debit or Credit transaction exists for the original pending-debit or pending-credit it is not possible any more.
- A request for a void transaction must contain a parent-transaction ID referring to the pending-debit or pending-credit transaction that needs to be voided.

7.6.1.1 Fields

The following elements are mandatory for sending a request/response for a void transaction for the payment method SEPA Direct Debit or SEPA Credit.

Some of the elements that are mandatory for a SEPA direct debit or SEPA credit transaction are not mandatory for a SEPA void transaction.

If the amount is sent within the request for the void transaction it will be checked if it is the same amount as in the previous pending-debit or pending-credit request. Partial cancellations are currently not possible.

Term	Man/Opt Req/Resp	Format	Max Size	Description
merchant-account-id	M (Req)	Alphanumeric	36	Unique identifier for a merchant account

requested-amount	O (Req)	Numeric	11,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency. Min amount €0.01. Max amount €999999999.99. If it is sent within the request it will be validated if it is the same amount like in the pending-debit or pending-credit request.
requested-currency	O (Req)	Alpha	3	This is the currency of the requested-amount. Only EUR (euro) is accepted.
request-id	M (Req)	Alphanumeric	150	This is the identification number of the request on the merchant's side. **It must be unique for each request.**
status.code	M (Resp)	Alphanumeric	12	This is the code of the status of a transaction
status.description	M (Resp)	Alphanumeric	256	This is the description to the status code of a transaction
status.severity	M (Resp)	Alphanumeric	20	This field gives information if a status is a warning, an error or an information
statuses.status	M (Resp)	Alphanumeric	12	This is the status of a transaction
transaction-id	M (Resp)	Alphanumeric	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
parent-transaction-id	M (req)	Alphanumeric	36	Transaction ID of the transaction that needs to be voided.

transaction-type	M (Req)	Alphanumeric	30	This is the type for a transaction. For a SEPA VOID request, only VOID-PENDING-DEBIT or VOID-DEBT or VOID-PENDING-CREDIT or VOID-CREDIT are allowed.

7.6.1.2 XML Samples for SEPA Void Transactions

Successful Void Pending-Debit Request

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <payment-methods>
        <payment-method name="sepadirectdebit"/>
        </payment-methods>
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
          account-id>
        <request-id>f91d0d2c-a77b-4270-a091-c9daea5cd32b</request-id>
        <transaction-type>void-pending-debit</transaction-type>
        <parent-transaction-id>3f8e01bc-9203-11e2-abbd-005056a96a54</parent-
          transaction-id>
        <requested-amount currency="EUR">10.00</requested-amount>
         </payment>
```

Successful Void Pending-Debit Response (for the previous request)

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
        <transaction-id>80538a32-920f-11e2-97cb-d4bed99f2247</transaction-id>
        <request-id>f91d0d2c-a77b-4270-a091-c9daea5cd32b</request-id>
        <transaction-type>void-pending-debit</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2013-03-21T11:10:07.829+01:00</completion-time-stamp>
        <statuses>
        <status code="200.0000" description="Request successful" severity="information" The
request
          completed successfully/>
        </statuses>
        <requested-amount currency="EUR">10.00</requested-amount>
        <payment-methods>
        <payment-method name="sepadirectdebit"/>
        </payment-methods>
        </payment>
```

Failed Void Pending-Debit Request (Parent transaction is missing)

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <payment-methods>
        <payment-method name="sepadirectdebit"/>
        </payment-methods>
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
          account-id>
        <request-id>f91d0d2c-a77b-4270-a091-c9daea5cd32b</request-id>
        <transaction-type>void-pending-debit</transaction-type>
        <requested-amount currency="EUR">10.00</requested-amount>
         </payment>
```

Failed Void Pending-Debit Response (for the previous request)

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
        <transaction-id>80538a32-920f-11e2-97cb-d4bed99f2247</transaction-id>
        <request-id>f91d0d2c-a77b-4270-a091-c9daea5cd32b</request-id>
        <transaction-type>void-pending-debit</transaction-type>
        <transaction-state>failed</transaction-state>
        <completion-time-stamp>2013-03-21T11:10:07.829+01:00</completion-time-stamp>
        <statuses>
                <status code="400.1021" description= "The Parent Transaction Id is required,
        and not provided. Please check your input and try again.." severity="error"/>
        </statuses>
        <requested-amount currency="EUR">10.00</requested-amount>
        <payment-methods>
        <payment-method name="sepadirectdebit"/>
        </payment-methods>
        </payment>
```

Successful Void Pending-Credit Request

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <payment-methods>
        <payment-method name="sepacredit"/>
        </payment-methods>
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
          account-id>
        <request-id>e91d0d2c-a77b-4270-a091-c9daea5cd32a</request-id>
        <transaction-type>void-pending-credit</transaction-type>
        <parent-transaction-id>3f8e01bc-9203-11e2-abbd-005056a96a54</parent-
          transaction-id>
        <requested-amount currency="EUR">10.00</requested-amount>
         </payment>
```

Successful Void Pending-Credit Response (for the previous request)

```
        <payment xmlns="http://www.elastic-payments.com/schema/payment">
        <merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
        <transaction-id>80538a32-920f-11e2-97cb-d4bed99f2247</transaction-id>
        <request-id>e91d0d2c-a77b-4270-a091-c9daea5cd32a</request-id>
        <transaction-type>void-pending-credit</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2013-03-21T11:10:07.829+01:00</completion-time-stamp>
        <statuses>
        <status code="200.0000" description="Request successful" severity="information" The
request
          completed successfully/>
        </statuses>
        <requested-amount currency="EUR">10.00</requested-amount>
        <payment-methods>
        <payment-method name="sepacredit"/>
        </payment-methods>
        </payment>
```

Failed Void Pending-Credit Request (Parent transaction is missing)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<payment-methods>
<payment-method name="sepadirectcredit"/>
</payment-methods>
<merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-
  account-id>
<request-id>g91d0d2c-a77b-4270-a091-c9daea5cd32c</request-id>
<transaction-type>void-pending-credit</transaction-type>
<requested-amount currency="EUR">10.00</requested-amount>
 </payment>
```

Failed Void Pending-Credit Response (for the previous request)

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>eefc804c-f9d3-43a8-bd15-a1c92de10000</merchant-account-id>
<transaction-id>80538a32-920f-11e2-97cb-d4bed99f2247</transaction-id>
<request-id>g91d0d2c-a77b-4270-a091-c9daea5cd32c</request-id>
<transaction-type>void-pending-credit</transaction-type>
<transaction-state>failed</transaction-state>
<completion-time-stamp>2013-03-21T11:10:07.829+01:00</completion-time-stamp>
<statuses>
        <status code="400.1021" description= "The Parent Transaction Id is required,
and not provided. Please check your input and try again." severity="error"/>
</statuses>
<requested-amount currency="EUR">10.00</requested-amount>
<payment-methods>
<payment-method name="sepacredit"/>
</payment-methods>
</payment>
```

7.7.1 Deposits

Occasionally, an end-customer send funds back to the merchant, either at the request of the merchant or through his own initiative. There are several reasons for this such as a re-payment or a subsequent / supplementary payment for an already existing SEPA Direct Debit Transaction.

These supplementary payments enter the Wirecard Payment Processing System as transactions with transaction type “Deposit” and with the payment method “wire transfer”.

If the end-customer provides the PTRID (Provider Transaction Reference ID) in the descriptor of his transfer, the Wirecard Payment Processing System will attempt to match the Deposit with the original SEPA Direct Debit or SEPA Credit transaction. If the PTRID is not provided or it is simply not possible to match the transaction the transaction will remain unmatched.

7.8.1 Bank Data Validations

To ensure transactions with faulty bank data are not forwarded to the bank for processing, the Wirecard Payments Platform validates the consumer’s bank data.

7.8.1.1 IBAN Validation

The consumer’s IBAN is validated in the following manner:

- The validity of the country code within the IBAN
- The validity of the country code within the SEPA scheme
- The structure of the IBAN including:
    - The length of the IBAN based on the country of issue
    - Position of the bank identifier
    - Position of the country code
    - Position of the check digits
- Check digit verification

7.8.1.2 BIC Validation

The consumer’s BIC is validated in the following manner:

- The length of the BIC may only be either 8 or 11 characters
- The validity of the BIC within the SEPA scheme

7.8.1.3 IBAN Only

The IBAN Only functionality allows a transaction to be processed without needing to submit a BIC. This is currently only possible for transactions whereby the country of the merchant’s IBAN is identical to the country of the consumer’s IBAN (ie: *national* transactions).

If a merchant submits a transaction without a BIC, a validation will pass to ensure the transaction really may be processed as IBAN Only.

For example:

Merchant Bank Account: **DE**11222222223333333333
Account Holder Bank Account: **DE**22333333334444444444
BIC NOT REQUIRED

Merchant Bank Account: **DE**11222222223333333333
Account Holder Bank Account: **AT**556666677777777777
BIC REQUIRED

For most Wirecard merchants, this means that ‘IBAN Only’ will be available exclusively for Germany.

10 Appendix A: Payment Methods

Id	Name	Description
Banktransfer	Bank Transfer	A method of transferring money from one person or institution (entity) to another, via bank accounts.
boleto_bancario	Boleto Bancario	Boleto Bancario is one of the most trusted online payment methods in Brazil. Upon selecting Boleto Bancario as their preferred payment method on the merchant's website, a "Boleto" a pre-filled bank slip including the purchase details is generated. The consumer then has the choice to either print the Boleto and pay by cash at a participating kiosk or to pay via his online banking service.
carrier-billing	Carrier Billing	A payment method using the SMS service provided by mobile network operators (carriers).
Cimbclicks	CIMB Clicks	CIMB Clicks (formerly known as Channel-e) is the Internet banking and financial services portal accessible to all CIMB Bank's ATM and Credit Cardholders.
Clickandbuy	clickandbuy	An online Ewallet that faciliates online purchases.
Creditcard	Credit Card	A payment method using a plastic card issued by a bank, business, etc., for the purchase of goods or services on credit.
debito_bradesco	Debito Bradesco	Debito Bradesco is an online banking payment method offered by Banco Bradesco Brazil, one of the largest banks in Brazil. It allows consumers that are customers of Banco Bradesco to pay online using their bank account. The consumer simply needs to enter his Banco Bradesco debit card number and is transferred to his online banking page where he confirms the transaction with a PIN number.
direct_transfer	Direktüberweisung	Direktüberweisung is an online banking payment method offered by Telekom. It allows consumers to pay online using their bank account. The consumer simply needs to enter his Bank Account Data and has to transfer the transaction with a PIN number.
EFT	Electronic Fund Transfer	A payment method based on a system of transferring money from one bank account directly to another without any paper money changing hands.
eplatby	ePlatby	ePlatby (ePlatby for eKonto or just eKonto) is an online bank transfer solution offered by Raiffeisenbank in the Czech Republic.

Eps	EPS	The eps e-payment standard is the interface into online payment systems of Austrian Banks for irrevocable payments, adopted by Internet-Shops as well as public authorities like E-Government.
Euteller	EUTeller	An Electronic Fund Transfer for Finnish Customers.
Gluepay	GluePay	An Electronic Fund Transfer for Sweden Customers.
Ideal	iDEAL	A mechanism by which customers to buy securely on the Internet using direct online transfers from their bank account in Netherland.
masterpass	MasterPass	MasterPass is a simple digital wallet for faster safe shopping, allowing to make purchases without entering shipping, and credit card information.
maybank2u	Maybank2u	A proprietary Maybank method of transferring funds and paying bills.
mercado_pago	Mercado Pago	MercadoPago is an online payment platform in Latin America that works like a wallet.
Mistercash	mistercash	An Electronic Fund Transfer for Mistercash Customers.
moneta_ru	Moneta.ru	Moneta.ru is a wallets in Russia and Ukraine.
p24	Przelewy24	An Electronic Fund Transfer for Polish Customers.
paypal	PayPal	A global e-commerce business allowing payments and money transfers to be made through the Internet.
Poli	POLi	An Electronic Fund Transfer for Australia Customers.
ratepay-elv	Guaranteed Direct Debit	Guaranteed Direct Debit allows customers shop online and the invoice amount due will be debited from their account afterwards.
ratepay-invoice	Guaranteed Invoice	Guarantede Invoice is comfortable and safe for online shoppers: receiving goods before paying.
Sepacredit	SEPA Credit Transfer	Credit Transfer payment method for Single Euro Payments Area
Sepadirectdebit	SEPA Direct Debit	Direct Debit payment method for Single Euro Payments Area
skrill	skrill	A global e-commerce business allowing payments and money transfers to be made through the Internet.
Sofortbanking	sofortbanking	sofortbanking is the direct payment method of Payment Network AG. sofortbanking allows you to directly and automatically trigger a credit transfer during your online purchase with your online banking information. A transfer order is instantly confirmed to merchant allowing an instant delivery of goods and services.
Transferencia	Transferencia entre contas	Transferencia entre contas is an online banking method in Brazil offered by Banco Bradesco Brazil
wiretransfer	Wire Transfer	Money transfer originating from and end-consumer

		and ending at a merchant
Yandex	Yandex	Yandex is one of Moneta.ru's funding options.

11 Appendix B: Transaction Types

Transaction Type Id	Name	Description
Authorization	Authorization	Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture] on this transaction.
authorization-supplementary	Authorization Supplementary	Reserves additional funds from the [Cardholder]'s account following an [Authorization]. 7 days left to conduct a [Capture] on this transaction.
authorization-only	Authorization Only	Creates billing agreement only for future reference [Debit]s and [Authorization]s.
capture-authorization	Capture	Takes funds from the [Card Holder]'s account. Must follow an [Authorization] or [Authorization Supplementary] chain.
check-enrollment	3-D Check Enrollment	Verifies the 3-D participation of a card using card scheme directory server and/or ACS.
check-payer-response	3-D Check Payer Response	Verifies that the cardholder authenticated the transaction within the 3-D Secure Program.
check-signature	Check Signature	Validates digital signature for a payment page transaction
Credit	Credit	Moves funds from the merchant to the Account Holder.
credit-return	credit-return	The forced reversal of a Credit Transaction because the bank can’t process the Credit transaction due to different reasons.
Debit	Debit	Bank transfer
debit-return	debit-return	The forced reversal of a Debit Transaction because the bank can’t process the Debit transaction due to different reasons.
Deposit	Deposit	An incoming payment from an end-customer to a merchant.
Detokenize	Detokenize	Provides sensitive Credit Card Data from a unique [Card Token].
first-cb-rep	First Chargeback Representment	First Chargeback Representment
get-url	Get URL	A step in a group of transactions where the provider or bank's URL is retrieved for the Account Holder to be redirected to

original-credit	Original Credit	Gives funds to the [Cardholder]'s account, NOT referring to an eligible [Purchase] or [Capture].
pending-debit	Pending Debit	Represents the request for a debit. If the [Transaction State] is success, then the outcome of the debit is not yet known. A further [Transaction Type] of debit follows, whose [Transaction State] indicates the outcome. If the [Transaction State] of the pending-debit is failed, then the request has not been processed.
pending-credit	pending-credit	Represents the request for a credit. If the [Transaction State] is success, then the outcome of the debit is not yet known. A further [Transaction Type] of credit follows, whose [Transaction State] indicates the outcome. If the [Transaction State] of the pending-credit is failed, then the request has not been processed.
Purchase	Purchase	Takes funds from the [Card Holder]'s account. A one-step process to conduct two Transaction Types: [Authorization] and [Capture].
referenced-authorization	Referenced Authorization	Reserve funds from the [Card Holder]'s account. Identical to a [Authorization] except for the fact that it Refers to a previous [Authorization] transaction.
referenced-purchase	Referenced Purchase	Takes funds from the [Card Holder]'s account. Identical to a [Purchase] except for the fact that it Refers to a previous [Purchase] transaction.
refund-capture	Refund Capture	Gives funds to the [Card Holder]'s account.
refund-purchase	Refund Purchase	Gives funds to the [Card Holder]'s account.
second-cb-rep	Second Chargeback Representment	Second Chargeback Representment
Tokenize	Tokenize	Provides a unique [Card Token] from sensitive Credit Card Data. The [Card Token] can be used in subsequent transactions instead of the card information itself.
void-authorization	Void Authorization	Frees and reserved funds from the [Card Holder]'s account due to an [Authorization] or chain of [Authorization Supplementary]s.
void-auth-supplementary	Voids authorization supplementary for CC	Voids an upwardly adjustment of an existing authorization.
void-capture	Void Capture	Frees reserved funds from the [Card Holder]'s account due to a [Capture].
void-credit	Void Credit	Frees reserved funds from the [Card Holder]'s account due to a [Credit].

void-original-credit	Void Original Credit	Frees reserved funds from the [Card Holder]'s account due to a [Orginal Credit].
void-purchase	Void Purchase	Frees reserved funds from the [Card Holder]'s account due to a [Purchase].
void-refund	Void Refund	Frees reserved funds from the [Card Holder]'s account due to an [Refund].
void-refund-capture	Void Refund Capture	Frees reserved funds from the [Card Holder]'s account due to an [Refund Capture].
void-refund-purchase	Void Refund Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Purchase].

Credit Card Transaction Types	
Authorization	✓
authorization-supplementary	✓
capture-authorization	✓
check-enrollment	✓
check-payer-response	✓
check-signature	✓
check-payer-response	✓
Detokenize	✓
first-cb	✓
first-cb-rep	✓
first-cb-succ-rep	✓
first-cb-unsucc-rep	✓
get-url	✓
original-credit	✓
purchase	✓
referenced-authorization	✓
referenced-purchase	✓
refund-capture	✓
refund-purchase	✓
retrieval-request	✓
retrieval-request-rep	✓
second-cb	✓
second-cb-rep	✓
second-cb-succ-rep	✓
second-cb-unsucc-rep	✓
void-auth-supplementary	✓
tokenize	✓
void-authorization	✓
void-capture	✓
void-credit	✓
void-original-credit	✓
void-purchase	✓
void-refund	✓
void-refund-capture	✓
void-refund-purchase	✓

		Transaction Types													
		authorization	capture-authorization	check-signature	credit	debit	get-url	pending-debit	pending-credit	refund-capture	refund-debit	void-authorization	debit-return	credit-return	authorization-only
Alternative Payment Methods	boleto_bancario			✓	✓	✓	✓	✓							
	cimbclicks			✓		✓	✓								
	debito_bradesco			✓	✓	✓	✓	✓							
	eft					✓	✓								
	ideal			✓		✓	✓								
	maybank2u			✓		✓	✓								
	mercado_pago			✓	✓	✓	✓	✓							
	moneta_ru			✓	✓	✓	✓	✓							
	paypal	✓	✓	✓		✓	✓			✓	✓	✓			✓
	sofortbanking			✓		✓	✓								
	transferencia			✓	✓	✓	✓	✓							
	yandex			✓	✓	✓	✓	✓							
	direktüberweisung					✓	✓								
	SEPA	✓			✓	✓			✓	✓			✓	✓	
	Skrill Wallet				✓	✓	✓								
	eplatby			✓		✓	✓								
	ratepay-invoice	✓	✓	✓	✓					✓		✓			
	ratepay-elv	✓	✓	✓	✓					✓		✓			
	masterpass			✓		✓	✓								

12 Appendix C: Transaction States

Transaction State	Description
Failed	Transaction has experienced an outcome other than the desired one.
in-progress	Transaction processing has begun.
Success	Transaction has been completed with a positive outcome.

13 Appendix D: Card Types

Card Type	Name
Amex	American Express
Diners	Diners Club
Discover	Discover
Jcb	JCB
Maestro	Maestro
Mastercard	Mastercard
Uatp	UATP
Visa	Visa

14 Appendix E: Transaction Statuses

Severity	Status Code	Name	Description
Information	100.0000	Transaction In Progress	Transaction Processing has begun, and the outcome of the Transaction is not yet known.
Information	100.0001	Data Reconciliation Matches	The data matches with a reconciling data source.
Information	100.0002	Data Reconciliation Mismatch	The data does not match with a reconciling data source.
Information	100.0003	Amount Reconciliation Mismatch	The transaction amount does not match with a reconciling data source.
Information	100.0004	State Reconciliation Mismatch	The transaction state (success/fail) does not match with a reconciling data source.
Information	100.5555	Test Transaction	Test Transaction
Information	200.0000	Request successful	The request completed successfully.
Information	200.1078	Successful with 3-D Full Authentication	The payment was successful, and the 3-D Secure Authentication occurred.
Information	200.1079	Successful with 3-D Attempted Authentication	The payment was successful, and the 3-D Secure Authentication was attempted, but not confirmed or denied.
Information	200.1080	Successful with 3-D Failed Authentication	The payment was successful, and the 3-D Secure Authentication failed.
Information	200.1081	Successful with 3-D Authentication Error	The payment was successful, and the 3-D Secure Authentication had an error and could not be confirmed.
Information	200.1083	Cardholder Successfully authenticated.	Cardholder Successfully authenticated.
Information	200.1084	Proof of authentication attempt was generated.	Proof of authentication attempt was generated.
Information	201.0000	Creation Success	The resource was successfully created.
Information	201.1125	Successful notification received from the bank.	Successful notification received from the bank.
Information	201.1126	Successful confirmation received from the bank.	Successful confirmation received from the bank.
Information	201.1127	Successful confirmation received from the third party	Successful confirmation received from the third party
Information	400.1193	Cancellation period expired	Cancellation period expired.
Information	409.1075	Lock Version Conflict (Delete)	Request cannot be processed due to a lock version conflict in a deletion operation. Please refresh your input and try again.
Information	409.2000	Aggregated Transaction	The transaction will be processed as an aggregated transaction.
Information	500.1075	Proof of authentication attempt was generated.	The payment reached the 3-D secure attempted status. The cardholder is not participating, but the attempt to authenticate was recorded. The

Severity	Status Code	Name	Description
			transaction reached the liability shift and payment can be accepted.
Information	501.1088	Requested Function Not Supported	The requested function is not supported.
Warning	100.0005	Disputed Transaction ID not located	The Disputed Transaction was not located, but the Provider Account was located.
Warning	100.0006	Disputed Transaction failed	The Disputed Transaction has the state of failed, the dispute record is still applied.
Warning	100.0007	Disputed Transaction Amount too high	The Disputed amount is higher than the Transaction's Funded Amount. (Funded Amount is the amount due to the merchant).
Warning	100.0008	Disputed Transaction Account Number Mismatch	The Disputed Account Number does not match, but the Provider Transaction ID does match.
Warning	400.1005	No Card Type	The Card Type has not been provided.
Warning	400.1032	No Chargeback Id	The Chargeback Id has not been provided.
Warning	400.1033	No Chargeback Date	The Chargeback Date has not been provided.
Warning	400.1034	No Chargeback Amount	The Chargeback Date has not been provided.
Warning	400.1035	No Chargeback Reason Code	The Chargeback Reason Code has not been provided.
Warning	400.1036	No Chargeback Provider Transaction Id	The Chargeback Provider Transaction Id has not been provided.
Warning	400.1037	No Chargeback Provider Account	The Chargeback Provider Account' has not been provided.
Warning	400.1038	Provider Transaction Id not found	The Provider Transaction Id does not exist.
Warning	400.1039	Duplicate Chargeback	This chargeback has already been posted. This is another chargeback being posted.
Warning	400.1040	Disputed Transaction already Refunded	The disputed transaction has already been refunded.
Warning	400.1041	Chargeback flow is misordered	Chargeback flow is misordered. An expected previous chargeback is missing.
Warning	400.1042	Different Chargeback Currency	Chargeback is in a different currency than the disputed transaction.
Warning	400.1043	Different Chargeback Amount	Chargeback amount is different than the disputed transaction.
Warning	400.1044	Chargeback over 6 months old	The Chargeback Date is more than 6 months after the disputed transaction.
Warning	400.1045	Chargeback Account Number mismatch	The Chargeback Account Number's last 4 digits do not match the Disputed Transaction.
Warning	400.1166	Duplicate Dispute Record	This record was already disputed.
Warning	400.1167	Duplicate Reconciliation Record	This record was already reconciled.
Warning	400.1171	Parent transaction consumer-id mismatch	The Consumer-Id of the Parent Transaction ID does not match the Consumer-Id of the current operation. Please check your input and try again.
Warning	400.1172	Parent transaction risk-reference-id mismatch	The Risk-Reference-Id of the Parent Transaction ID does not match the Risk-Reference-Id of the

Severity	Status Code	Name	Description
			current operation. Please check your input and try again.
Warning	400.1173	Parent transaction order-number mismatch	The Order-Number of the Parent Transaction ID does not match the Order-Number of the current operation. Please check your input and try again.
Warning	400.1178	Unable to confirm card type from card number	Unable to confirm card type from card number
Warning	400.1190	Malformated timeout string	Malformated timeout string
Warning	400.1191	Unrecognized requested status code	Unrecognized status code
Warning	500.1072	Card Not Enrolled	Card not enrolled: The card is not enrolled / the cardholder is not participating in the 3-D Secure program.
Warning	500.1073	Unable to Verify Enrolment	Issuer unavailable: The card issuing system is not reachable. The 3-D secure process cannot be verified.
Warning	500.1074	MPI Error	System unavailable: The MPI system is not reachable. The 3-D secure process cannot be verified.
Warning	500.1999	Acquirer: Unknown response	The acquirer returned an unknown response. Contact Technical Support.
Error	400.1000	Invalid Account Number (LUHN)	LUHN Check failed on the credit card number.
Error	400.1001	No Account Number	The Account Number has not been provided.
Error	400.1002	Invalid Account Number Length	The Account Number does not have the correct length.
Error	400.1003	Invalid Account Number Expiry Month	The Expiry month is invalid.
Error	400.1004	Invalid Account Number Expiry Year	The Expiry Year is invalid.
Error	400.1006	No Card Security Code	The Card Security Code has not been provided.
Error	400.1007	No Account Holder	The account holder information has not been provided.
Error	400.1008	No Merchant Account Identifier	The Merchant Account Identifier has not been provided.
Error	400.1009	No Transaction Type	The Transaction Type information has not been provided.
Error	400.1010	No Request Id	The Request Identifier has not been provided.
Error	400.1011	No Requested Amount	The Requested Amount has not been provided.
Error	400.1012	No Requested Amount Currency	The Requested Amount Currency has not been provided.
Error	400.1013	Requested Amount Below Minimum	The Requested Amount is below the minimum required for this Merchant Account.
Error	400.1014	No First Name	The First Name has not been provided.
Error	400.1015	No Last Name	The Last Name has not been provided.
Error	400.1016	Invalid Email	The Email Address is syntactically incorrect.

Severity	Status Code	Name	Description
Error	400.1017	Account Number not Numeric	The Account Number is not numeric; it should only have digits.
Error	400.1018	Duplicate Request Id	The same Request Id for the Merchant Account is being tried a second time. Use another Request Id.
Error	400.1019	Unrecognized Merchant Account Id	This Merchant Account Identifier does not exist, or is not assigned to this Processing User.
Error	400.1020	Unrecognized Parent Transaction Id	The Parent Transaction Id does not exist.
Error	400.1021	No Parent Transaction Id	The Parent Transaction Id is required, and not provided.
Error	400.1022	Parent Transaction Id not Successful	The Parent Transaction Id was not successful; the operation is not possible.
Error	400.1023	Parent Transaction Transaction Type not Valid for Current Request	The Transaction Type of the Parent Transaction is invalid for the current operation.
Error	400.1024	Invalid Parent Transaction Id	The Parent Transaction Id is invalid.
Error	400.1025	Parent Transaction Merchant Account Mismatch	The Parent Transaction Id does not exist for this Merchant Account Identifier.
Error	400.1026	Parent transaction currency mismatch	The Currency of the Parent Transaction ID does not match the Currency of the current operation.
Error	400.1027	Amount threshold exceeded for the parent transaction	The Requested Amount exceeds the Parent Transaction Amount.
Error	400.1028	Card token or account number is required for the transaction	The Card Token or Account Number has not been provided.
Error	400.1029	Card token does not match with the masked card number	The Card Token does not match the masked Account Number.
Error	400.1030	Invalid Currency	The Currency is invalid.
Error	400.1031	Malformed Request	Malformed request. Syntax of the request is invalid.
Error	400.1046	Invalid Country Code	The Country Code is invalid.
Error	400.1047	Invalid Request Id	The Request Id is greater than 36 characters in length.
Error	400.1048	No Provider Account	This Merchant Account does not have a Provider Account associated with it.
Error	400.1049	Bad Credential	Invalid username and/or password
Error	400.1050	Chargeback transaction on 3-D Secure	Chargeback transaction on 3-D Secure.
Error	400.1051	Order Number Existed	Order Number has already been paid.
Error	400.1052	Currency not supported	Currency is not supported
Error	400.1072	No provider route	No providers found for the account. Merchant Account is not configured properly.
Error	400.1079	Failed to synchronize	Failed to synchronize with Tokenization(remote)

Severity	Status Code	Name	Description
		with Tokenization(remote) Server	Server
Error	400.1081	No Bank Account IBAN	The Bank Account IBAN information has not been provided.
Error	400.1082	No Bank Account BIC	The Bank Account BIC information has not been provided.
Error	400.1083	No Mandate ID	The Mandate ID information has not been provided.
Error	400.1084	Mandate ID Invalid	The Mandate ID is invalid.
Error	400.1085	No Mandate Signed Date	The Mandate Signed Date information has not been provided.
Error	400.1086	No Mandate Signed City	The Mandate Signed City information has not been provided.
Error	400.1087	No Mandate Signature Image	The Mandate Signature Image information has not been provided.
Error	400.1098	Invalid Creditor ID	The Creditor ID is invalid.
Error	400.1099	Payment method URL building failed	Unable to resolve the payment method or URL building for payment method failed
Error	400.1100	Duplicate notification received from the bank	Duplicate notification received from the bank
Error	400.1102	Digital signature validation failed.	Digital signature validation failed.
Error	400.1103	Nonexistent Record	Update failed due to nonexistent record
Error	400.1105	Invalid Credentials	Username and/or Password contains a non ASCII characters
Error	400.1106	Duplicate Parent Transaction Id found	Duplicate Parent Transaction Id found
Error	400.1107	Duplicate Velocity Rule	Created Failed. Duplicate Merchant Account, Payment Method, Card Type, Currency, Measure Code, Time Period Code, Measure Action Code, Velocity Type Code, Single Card Flag
Error	400.1108	Duplicate Sequence Number	Created Failed. Duplicate Merchant Account, Sequence Number
Error	400.1109	Invalid Alternative Payment Method	Invalid Payment Method
Error	400.1110	Invalid Bank Account Data	Use either Bank Account and Bank Code or IBAN and BIC. Please check your input and try again.
Error	400.1111	IBAN or BIC too long	IBAN or BIC are too long. Please check your input and try again.
Error	400.1112	Bank Account or Bank Code too long	Bank Account or Bank Code are too long. Please check your input and try again.
Error	400.1113	A value exceeds the allowed size	The transaction could not be processed because a value of a field is too long. Please check your input and try again.
Error	400.1114	Requested Amount Above Maximum	The Requested Amount is above the maximim required for this Merchant Account.
Error	400.1119	Velocity: Velocity Limit	The transaction was refused because this

Severity	Status Code	Name	Description
		Reached, Merchant Account	merchant account has reached a velocity limit. Contact technical support to adjust limits or wait and try again later.
Error	400.1120	Velocity: Velocity Limit Reached, Payer Account	The transaction was refused because the account paying has reached a limit. Contact technical support to adjust limits or wait and try again later.
Error	400.1121	Velocity: Transaction is below minimum amount.	The transaction was refused because the amount is too low. Contact technical support to adjust limits or try another amount.
Error	400.1122	Velocity: Transaction is above maximum amount.	The transaction was refused because the amount is too high. Contact technical support to adjust limits or try another amount.
Error	400.1123	Request timestamp (UTC) cannot be blank or empty.	The transaction was refused because signature expiry cannot be checked. Please check your input and try again.
Error	400.1124	Signature is already expired.	The transaction was refused because signature is expired. Ensure that the Request timestamp is in UTC, and that the timeout period has not passed. Try again if too much time has passed.
Error	400.1125	No Processing Redirect URL	The Processing Redirect URL information has not been provided. Please check your input and try again.
Error	400.1126	No IP Address	The IP Address information has not been provided.
Error	400.1127	Parent Amount Mismatch	The Transaction Amount does not qualify to the parent transaction amount. Please try another amount.
Error	400.1128	Missing 3D Element	The Current transaction must have 3D element and a check-enrollment parent transaction id.
Error	400.1129	Invalid descriptor-length	The descriptor is too long. Please check.
Error	400.1130	No expected fractional digits in the amount	The amount has not the expected fractional digits. Please check.
Error	400.1131	Invalid processing URL length	The processing URL is too long. Please check.
Error	400.1132	Invalid order number length	The order number is too long. Please check.
Error	400.1133	Plugin Validation error	Plugin Validation error.
Error	400.1134	No Provider Merchant Account	The Provider Merchant Account information has not been provided. Please check your input and try again.
Error	400.1135	No Username	The Username information has not been provided. Please check your input and try again.
Error	400.1136	No Password	The Password information has not been provided. Please check your input and try again.
Error	400.1137	No Provider Callback URL	The Provider Callback URL information has not been provided. Please check your input and try again.
Error	400.1138	No Payment Success Redirect URL	The Payment Success Redirect URL information has not been provided. Please check your input and try again.

Severity	Status Code	Name	Description
Error	400.1139	No Payment Cancel Redirect URL	The Payment Cancel Redirect URL information has not been provided. Please check your input and try again.
Error	400.1140	No Payment Fail Redirect URL	The Payment Fail Redirect information has not been provided. Please check your input and try again.
Error	400.1141	Parent Card Mismatch	The transaction card info does not match the parent transaction card info. Please check your input and try again.
Error	400.1142	No Payment Cancel Redirect URL	No Payment Cancel Redirect URL information has not been provided. Please check your input and try again.
Error	400.1143	No Descriptor	The Descriptor has not been provided. Please check your input and try again.
Error	400.1144	Missing Mandatory Field	The request is missing a mandatory field. Please check your input and try again.
Error	400.1145	Invoice Number too long	Invoice Number is too long. Please check your input and try again.
Error	400.1146	Date format not correct	A date parameter has not the correct format. Please check your input and try again.
Error	400.1147	Track Data Account Number Mismatch	The track data account number does not match with the input account number. Please do not pass the account number when track data is provided.
Error	400.1148	Track Data Expiration Month Mismatch	The track data expiration month does not match with the input expiration month. Please do not pass the expiration month when track data is provided.
Error	400.1149	Track Data Expiration Year Mismatch	The track data expiration year does not match with the input expiration year. Please do not pass the expiration year when track data is provided.
Error	400.1150	Unsupported characters	Unsupported characters found in the request. Please check your input.
Error	400.1153	Unknown Merchant Account	The merchant account couldn't be resolved based on the data provided. This might be caused by incomplete data or missing configuration.
Error	400.1163	Invalid Bank Account Data	For foreign bank accounts only IBAN and BIC are allowed. Please adjust your input and try again
Error	400.1168	No Order Number	The Order number is required, but not provided. Please check your input and try again.
Error	400.1169	Invalid language	The language or language_COUNTRY is invalid
Error	400.1170	Invalid Periodic Type	The Periodic Type is invalid
Error	400.1174	Invalid Entry Mode	The Entry Mode is invalid.
Error	400.1175	Invalid Due Date	The Due Date is invalid.
Error	400.1176	Street1 is too long	The information submitted in Street1 is too long. Please re-submit.
Error	400.1177	IBAN and BIC countries do not match	The country codes of the IBAN and BIC are not the same. Please check your input and try again.
Error	400.1179	Mismatch card type and	Mismatch card type and card number

Severity	Status Code	Name	Description
		card number	
Error	400.1180	Request timestamp (GMT) is in future compared to time of the system (GMT).	The transaction was refused because signature expiry cannot be checked.
Error	400.1181	No personal identification number	The personal identification number has not been provided. Please check your input and try again.
Error	400.1182	No promotion code	The promotion code has not been provided. Please check your input and try again.
Error	400.1183	Invalid sequence type	The sequence type is invalid. Please check your input and try again.
Error	400.1184	Invalid Mandate Signed Date	The Mandate Signed Date is invalid.
Error	400.1185	Default transaction type not configured	There is no default auto-sale transaction configured for this payment method.
Error	400.1186	Order items overall amount is different to requested amount	Order items overall amount is different to requested amount. Please check your input and try again.
Error	400.1187	Every order item should have same currency as requested amount	Every order item should have same currency as requested amount. Please check your input and try again.
Error	400.1192	Wrong parent transaction test mode	Parent transaction is not a test Transaction
Error	400.1194	IBAN country code is invalid	IBAN country code is invalid.
Error	400.1195	IBAN is not SEPA compliant	IBAN is not SEPA compliant.
Error	400.1196	Invalid IBAN	IBAN is invalid.
Error	400.1197	Invalid BIC	BIC is invalid
Error	400.1198	BIC is not SEPA Compliant	BIC is not SEPA Compliant
Error	400.1199	Merchant account not properly configured	The Merchant Account is not properly configured for processing. Please contact technical support.
Error	400.1200	Parent transaction was cancelled by merchant	Parent transaction was cancelled by merchant.
Error	400.1201	No Email	Email has not been provided. Please check your input and try again.
Error	400.1202	No Phone	Phone has not been provided. Please check your input and try again.
Error	400.1203	No Date of birth	Date of birth has not been provided. Please check your input and try again.
Error	400.1204	No Street1	Street1 has not been provided. Please check your input and try again.
Error	400.1205	No City	City has not been provided. Please check your input and try again.
Error	400.1206	No Country	Country has not been provided. Please check your input and try again.

Severity	Status Code	Name	Description
Error	400.1207	No Postal code	Postal code has not been provided. Please check your input and try again.
Error	400.1208	No Order items	No Order items have been provided. Please check your input and try again.
Error	400.1209	No Order item name	Order item name has not been provided. Please check your input and try again.
Error	400.1210	No Order item article number	Order item article number has not been provided. Please check your input and try again.
Error	400.1211	No Order item amount	Order item amount has not been provided. Please check your input and try again.
Error	400.1212	No Order item quantity	Order item quantity has not been provided. Please check your input and try again.
Error	400.1213	Order item quantity is not valid	Order item quantity is not valid. Please check your input and try again.
Error	400.1214	Bank account missing.	Bank account missing.
Error	400.1215	Bank account data invalid.	Use either Bank Account and Bank Code or IBAN.
Error	400.1216	Bank name missing.	Bank name missing.
Error	400.1217	Bank name length invalid.	Bank name length invalid.
Error	400.1218	Bank code length invalid.	Bank code length invalid.
Error	400.1219	Bank code invalid.	Bank code invalid.
Error	400.1220	No Order item tax rate	Order item tax rate has not been provided. Please check your input and try again.
Error	400.1222	Ambiguous Order item tax	Ambiguous order item tax. Use either tax amount or tax rate only.
Error	400.1223	Invalid Order item tax rate	Order item tax rate is out of range. Please check your input and try again.
Error	403.1166	Access Denied	User doesn't have the access role for the requested operation.
Error	409.1073	Failed to create: Duplicate Card Type, Currency, Provider to Merchant Account	Creation Failed. Duplicate Card Type, Currency, Provider and Merchant Account
Error	409.1074	Lock Version Conflict (Update)	Request cannot be processed due to a lock version conflict in an update operation.
Error	409.1076	Failed to Create Provider Route	Creation Failed. Merchant Account Provider Route.
Error	409.1077	Failed to Update Provider Route	Update Failed. Merchant Account Provider Route.
Error	409.1078	Duplicate Username	Username already in use.
Error	409.1101	Failed to create: Duplicate Provider and Payment Method to Merchant Account	Creation Failed. Duplicate Provider, Payment Method and Merchant Account
Error	409.1104	Failed to create: Duplicate Airline Code and XPath Expression	Creation Failed. Duplicate Airline Code and XPath Expression. Please check your input and try again.
Error	409.1105	Acquirer: Too many	Provide one of merchant account id, merchant

Severity	Status Code	Name	Description
		merchant account identifiers supplied	account resolver category or parent transaction id
Error	500.1000	Tokenization Server Error	Error getting response from Tokenization(remote) Server. Contact Technical Support.
Error	500.1049	Provider Busy	Provider is busy. Try again later.
Error	500.1050	Provider System Error	Provider had a system error. Try again later.
Error	500.1051	Provider Rejected Transaction	A Provider refused to accept the transaction.
Error	500.1052	Provider Unavailable	A Provider is unavailable.
Error	500.1053	Acquirer: Declined	The acquirer returned Declined.
Error	500.1054	Acquirer: Pick Up Card	The acquirer returned Pick up card.
Error	500.1055	Acquirer: Call Card Center	The acquirer returned Call card center.
Error	500.1056	Acquirer: Issuer not Available	The acquirer returned Card center is not available.
Error	500.1057	Acquirer: Ineligible Transaction	The acquirer returned Error or Ineligible Transaction.
Error	500.1058	Acquirer: Insufficient Funds	The acquirer returned Insufficient funds.
Error	500.1059	Acquirer: Invalid CVV	The acquirer returned Invalid Card Security Code.
Error	500.1060	Acquirer: Denied Transaction Type	The acquirer returned Transaction Type not accepted.
Error	500.1061	Acquirer: Card Type not Supported	The card type is not processed by the authorization center
Error	500.1062	Acquirer: Expired Card	The acquirer returned Expired Card
Error	500.1063	Acquirer: Voice Authorization Required	The acquirer returned Call Voice-authorization number; Initialization Data
Error	500.1064	Acquirer: Expired Card	The acquirer returned Expired Card
Error	500.1065	Acquirer: Stolen Card	The acquirer returned Stolen Card
Error	500.1066	Acquirer: Restricted Card	The acquirer returned Restricted Card
Error	500.1067	Acquirer: Issuer temporarily not reachable	The acquirer returned Card issuer temporarily not reachable
Error	500.1068	Acquirer: Processing temporarily not possible	The acquirer returned Processing temporarily not possible
Error	500.1069	Acquirer: Referral	Referral. Transaction was declined but could be approved with a verbal authorization
Error	500.1070	Acquirer: Lost Card	The acquirer returned Lost Card.
Error	500.1071	Acquirer: AVS Check Failure	The acquirer returned AVS Check Failure.
Error	500.1076	Customer failed or Cancelled authentication.	The cardholder not only cancelled but even denied the payment process. No liability shift is granted and payment should not be processed as customer expressed his explicit denial.
Error	500.1077	Authentication could not be completed, due to technical or other problem	The 3-D systems are not reachable and authentication could not be performed. No liability shift is granted

Severity	Status Code	Name	Description
Error	500.1085	A system error prevented completion of authentication.	A system error prevented authentication from completing. The card can be accepted for payment but no authentication information will be passed to authorization processing and no liability shift will take place.
Error	500.1086	Acquirer: Missing Mandatory Field	The acquirer is missing a mandatory field.
Error	500.1087	Connection to provider timed out.	The connection to the provider was never made and timed out.
Error	500.1088	Requested Function Not Supported	Requested Function Not Supported. Please check your input and try again.
Error	500.1089	Acquirer: Refund not permitted	This acquirer does not support refunds.
Error	500.1091	Acquirer: Suspicion of Manipulation	Suspicion of Manipulation
Error	500.1092	Acquirer: Card not in authorizer's database.	Card not in authorized Database
Error	500.1093	Acquirer: Exceeds cash withdrawal floor limit.	Exceeds cash withdrawal floor limit.
Error	500.1094	Acquirer: Merchant Account not configured.	The Merchant Account is not properly configured for processing.
Error	500.1095	Acquirer: Not in compliance with security regulations.	Not in compliance with security regulations.
Error	500.1096	Acquirer: Count threshold exceeded.	Count threshold exceeded.
Error	500.1097	Acquirer: Pin failure limit reached.	Pin failure limit reached.
Error	500.1098	Acquirer: Credit restriction violation.	Credit restriction violation.
Error	500.1099	Acquirer: Transaction processing refused.	Transaction processing refused.
Error	500.1100	Acquirer: Card Number not permitted in Demo Mode.	Acquirer: Card Number not permitted in Demo Mode.
Error	500.1101	Acquirer: Clearing file generated	File generation was performed successfully
Error	500.1102	Acquirer: TX submitted to financial partner	File was sent to the financial partner (bank)
Error	500.1103	Acquirer: TX successfully reconciled	Transaction reconciliation successful
Error	500.1104	Acquirer: Duplicate payment or transaction	Payment or transaction was identified as duplicate
Error	500.1105	Acquirer: Limit constraint	The maximum amount of allowed bookings has been reached
Error	500.1106	Acquirer: Debtor IBAN Country Restriction (country restriction for debtor bank account	The debtor IBAN is not allowed based on country restriction

Severity	Status Code	Name	Description
		applies)	
Error	500.1107	Acquirer: Transaction Cancelled by Merchant	Transaction was cancelled by the merchant.
Error	500.1108	Acquirer: Transaction Cancelled	Transaction was cancelled.
Error	500.1109	Acquirer: Malformed/Invalid Parameter	Malformed/Invalid Parameter
Error	500.1110	Acquirer: Malformed/Invalid Signature	Malformed/Invalid Signature
Error	500.1111	Acquirer: Account status was not updated	Account status was not updated
Error	500.1112	Acquirer: Account does not exist	Account does not exist
Error	500.1113	Acquirer: Account already exists	Account already exists
Error	500.1114	Acquirer: Transaction is 3-D enrolled.	Transaction is 3-D enrolled.
Error	500.1115	Acquirer: Currency not enabled for this merchant account.	Currency not enabled for this merchant account.
Error	500.1116	Acquirer: Invalid Card	Invalid Card
Error	500.1117	Acquirer: Terminal ID Unknown	Terminal ID Unknown
Error	500.1118	Acquirer: Invalid Transaction	Invalid Transaction
Error	500.1119	Velocity: Velocity Limit Reached, Merchant Account	The transaction was refused because this merchant account has reached a velocity limit.
Error	500.1120	Velocity: Velocity Limit Reached, Payer Account	The transaction was refused because the account paying has reached a limit.
Error	500.1121	Velocity: Transaction is below minimum amount.	The transaction was refused because the amount is too low.
Error	500.1122	Velocity: Transaction is above maximum amount.	The transaction was refused because the amount is too high.
Error	500.1123	Transaction type not supported	Could not determine Operation subtype.
Error	500.1127	Failed confirmation received from the third party	Failed confirmation received from the third party.
Error	500.1151	Terminal not ready	Terminal not ready
Error	500.1152	Amount larger	The sum of the credited amount is larger than the original debit
Error	500.1154	Invalid Amount	The amount requested is not valid.
Error	500.1155	Invalid Country Code	The Country Code is invalid.
Error	500.1156	Issuer: Declined	The issuer returned Declined. Please check with

Failed confi

Severity	Status Code	Name	Description
			Issuer, or use different card.
Error	500.1157	Issuer: Stolen Card	The issuer returned Stolen Card. Please check with Issuer, or use different card.
Error	500.1158	Issuer: Busy, try again later	The issuer returned Processing temporarily not possible. Please try again later.
Error	500.1159	Issuer: Invalid Card	The issuer returned Invalid Card. Please check your input or use different card.
Error	500.1160	Issuer: Suspected Fraud	The issuer returned Suspected Fraud. Please check with Issuer, or use different card.
Error	500.1161	Issuer: Transaction Type not accepted	The issuer returned Transaction Type not accepted. Please contact technical support.
Error	500.1162	Issuer: Expired Card	The issuer returned Expired Card. Please check your input or use different card.
Error	500.1999	Unknown response	The acquirer returned an unknown response. Contact Technical Support.
Error	500.2100	Request processing failure.	Request processing failure.
Error	500.2200	No response from process within timeout settings.	No response from process within timeout settings.
Error	500.2376	Cancellation period expired.	Cancellation period expired.
Error	500.2377	Already settled referenced authorization found.	Already settled referenced authorization found.
Error	500.2378	Requested debit exceeds the available authorized fund.	Requested debit exceeds the available authorized fund.
Error	500.2379	Inconsistent referenced transaction information found.	Inconsistent referenced transaction information found.
Error	500.2380	Account blacklist check failure.	Account blacklist check failure.
Error	500.2390	Mandate not found.	Mandate not found.
Error	500.2391	Debit transaction cancelled.	Debit transaction cancelled.
Error	500.2392	Credit transaction cancelled.	Credit transaction cancelled.
Error	500.2395	Authorization transaction cancelled	Authorization transaction cancelled
Error	500.2396	Authorization transaction partially cancelled	Authorization transaction partially cancelled
Error	500.2400	Debtor account closed	Debtor account closed
Error	500.2401	insufficient funds/ no funds	insufficient funds/ no funds
Error	500.2402	Direct debit refuse by debtor/ payer disagree	Direct debit refuse by debtor/ payer disagree
Error	500.2403	No authorisation from	No authorisation from debtor to his bank

Severity	Status Code	Name	Description
		debtor to his bank	
Error	500.2409	Account blocked	Account blocked
Error	500.2414	not direct debited	not direct debited
Error	500.2416	Bank cancellation	Bank cancellation
Error	500.2417	unknown Account number	unknown Account number
Error	500.2418	Unusable Bank Code	Invalid Bank Code
Error	500.2419	unknown beneficiary	unknown beneficiary
Error	500.2420	unknown issuer	unknown issuer
Error	500.2421	claim yielded to other bank	claim yielded to other bank
Error	500.2422	no order to pay	no order to pay
Error	500.2436	account number unknown	account number unknown
Error	500.2442	debit not approved	debit not approved
Error	500.2443	double payment	double payment
Error	500.2453	direct debit not possible	direct debit not possible
Error	500.2454	credit payment not possible	credit payment not possible
Error	500.2494	cancelled on request	cancelled on request
Error	500.2495	cancellation order executed	cancellation order executed
Error	500.2496	cancelled instead of bended	cancelled instead of bended
Error	500.2497	Amount Too Low	The Requested Amount is below the minimum required.
Error	500.2498	Amount Too High	The Requested Amount is above the maximum required.
Error	500.2499	Authentication Error or Failure	The authentication attempt failed and/or had an error.
Error	500.2500	Failed Authentication pin/phone/password/etc	The Account Holder failed to authenticate with pin, password, phone number, or some other mechanism.
Error	500.2600	Undebit not possible.	Undebit not possible.
Error	500.2601	Partner ID not found.	Partner ID not found.
Error	500.2602	Promotion code is not valid.	Promotion code is not valid.
Error	500.2603	Provider account is not valid.	Provider account is not valid.
Error	500.2604	Max number of unsuccessful logins exceeded.	Max number of unsuccessful logins exceeded.
Error	500.2605	Requested version of function not supported	Requested version of function is not supported.
Error	500.2606	Session expired	Provider session has expired.
Error	500.3000	Risk: Rejected due to Suspicion of Fraud.	Rejected due to Suspicion of Fraud.
Error	500.3001	Risk: Insufficient data for the verification.	Error in transaction request; check request data

Severity	Status Code	Name	Description
Error	500.3002	Risk: General risk management rejection	Risk: General risk management rejection
Error	501.1090	Content Type not Supported.	The content type is not supported.
Error	999.9998	Database error	Database error
Error	999.9999	System error	System error

15 Appendix F: Countries and Currencies

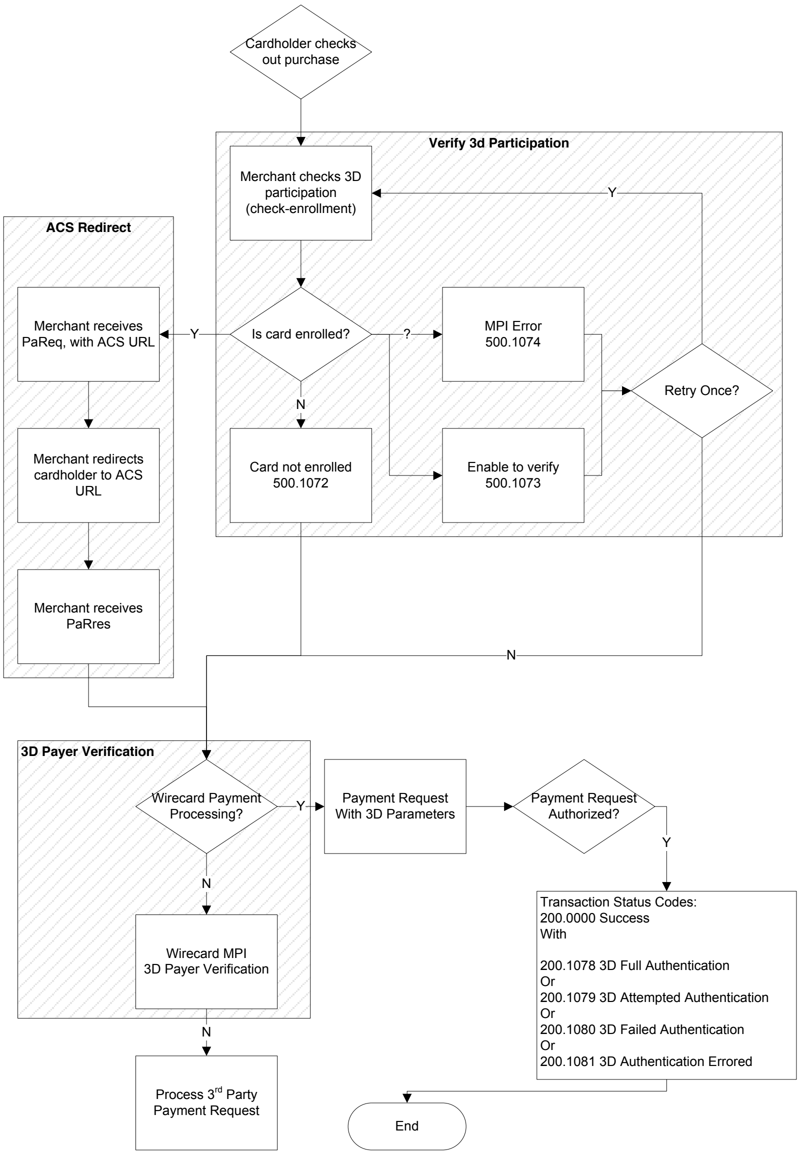
Countries follow the ISO 3166-1-alpha-2 code.

http://www.iso.org/iso/country_codes/iso_3166_code_lists/country_names_and_code_elements.htm

Currencies follow the ISO 4217 3-char alphabetic code.

http://www.iso.org/iso/currency_codes

16 Appendix G: 3-D-Secure Process Flow


Cardholder checks out purchase
Verify 3d Participation
Merchant checks 3D participation (check-enrollment)
Y
ACS Redirect
Merchant receives PaReq, with ACS URL
Y
Is card enrolled?
?
MPI Error 500.1074
N
Retry Once?
Merchant redirects cardholder to ACS URL
Card not enrolled 500.1072
Enable to verify 500.1073
Merchant receives PaRres
N
3D Payer Verification
Wirecard Payment Processing?
Y
Payment Request With 3D Parameters
Payment Request Authorized?
Y
N
Wirecard MPI 3D Payer Verification
N
Process 3rd Party Payment Request
End

Transaction Status Codes:
200.0000 Success
With

200.1078 3D Full Authentication
Or
200.1079 3D Attempted Authentication
Or
200.1080 3D Failed Authentication
Or
200.1081 3D Authentication Errored

17 Appendix H: Glossary

Term	XML Element	Definition
Access Control Server URL	three-d.acs-url	This is the issuer URL to where the merchant must direct the enrolment check request via the cardholder's browser. It is returned only in case the cardholder is enrolled in 3-D secure program.
Account Holder	account-holder	The person or entity that has been issued a credit, prepaid or debit card and is responsible for compliance with the rules of engagement as contracted with the card-issuing body. This information is provided in the card's terms and conditions.
Account Number	bank-account.account-number	The number designating a bank account used nationally
Bank Code	bank-account.bank-code	The national bank sorting code for national bank transfers
Cancel-Redirect-URL	payment.cancel-redirect-url	The URL to which the [Account Holder] will be re-directed after he has cancelled a payment. This is normally a page on the [Merchant]'s website.
Capture		Takes funds from the [Cardholder]'s account. Must follow an [Authorization].
Card Account Number	card.account-number	The embossed or encoded number that identifies the card issuer to which a transaction is to be routed and the account to which it is to be charged unless specific instructions indicate otherwise. In the case of a Credit Card, this is the Primary Account Number.
Cardholder Authentication Verification Value	three-d.cardholder-authentication-value	The CAVV is a a cryptographic value generated by the Issuer. For Visa transaction it is called CAVV (Cardholder Authentication Verification Value) for MasterCard it is either called Accountholder Authentication Value (AAV) or Universal Cardholder Authentication Field (UCAF).
Card Expiry Month	card.expiry-month	The 2-digit representation of the expiration month of the [Card Account].
Card Expiry Year	card.expiry-year	The 4-digit representation of the expiration year of the [Card Account].
Card Masked Account Number	card.masked-account-number	A code used to represent the [Card Masked Account].
Card Security Code	card.security-code	A security feature for credit or debit card transactions, providing increased protection against credit card fraud. The Card Security Code is located on the back of MasterCard, Visa and Discover credit or debit cards and is typically a separate group of 3 digits to the right of the signature strip. On American Express cards, the Card Security Code is a printed, not embossed, group of four digits on the front towards the right.
Card Token Id	card.token-id	A unique identifier assigned for every [Card Token]. This is a surrogate value for the Primary Account Number.
Card Type	card.card-type	A card scheme accepted by the [Processing System]. This includes physically issued cards.
Credit		Gives funds to the [Cardholder]'s account, referring to an eligible [Purchase] or [Capture].
Currency		A generally accepted form of money, including coins and paper notes, which is issued by a government and circulated

Term	XML Element	Definition
		within an economy. Used as a medium of exchange for goods and services, currency is the basis for trade. For more information see ISO 4217 (International standard for currency codes).
Eci	three-d.eci	In a 3-D Secure process, this indicates the status of the [VERes].
Fail Redirect URL	payment.fail-redirect-url	The URL to which the [Account Holder] will be re-directed after an unsuccessful payment. This is normally a page on the [Merchant]\'s website notifying the [Account Holder] of a failed payment often with the option to try another [Payment Method].
Merchant		A person or firm contractually affiliated with the [Payment Service Provider] for accepting payments for services rendered.
Merchant Account		Other terminology for [Merchant Processing Account].
Merchant Account Id	merchant-account-id	A unique identifier assigned for every [Merchant Account].
Merchant Data	three-d.md	A parameter in the 3-D Secure process reserved for specific merchant data. The MD may be useful for retrieving transaction data from the database or recalling a transaction.
Parent Transaction		The [Transaction] that precedes the current [Transaction] in a [Transaction Group]. For example, a [Capture] Transaction must have an [Authorization] Transaction as its parent.
PaReq	three-d.pareq	In a [3-D Secure] transaction, this is a base64-encoded request message created for cards participating in the 3-D program. The PaReq is returned by the issuer's ACS via the card schemes' directory servers to the Wirecard system and from here passed on to the merchant.
PARes	three-d.pares	In a [3-D Secure] transaction, this is the digitally signed, base64-encoded authentication response message received from the issuer.
Payment Service Provider		A company that offers [Merchant]s online services for accepting electronic payments by a variety of payment methods . The online services are provided by the [Operating Company].
Processing Redirect URL	payment.processing-redirect-url	The URL to which the [Account Holder] will be re-directed after an unsuccessful payment. This is normally a page on the [Merchant]'s website notifying the [Account Holder] of a failed payment.
Purchase		A transaction type initiated by a consumer or merchant which combines both a Pre-Auth and a Completion The Purchase transaction verifies funds on the customer's card, removes the funds and readies them for deposit into the merchant's account. Purchase combines both and Auth and a Capture into one transaction (or both a Pre-auth and a Completion). Known as one-step transaction processing.
Refund		Gives funds to the [Cardholder]'s account, WITHOUT referring to an eligible [Purchase] or [Capture].
Term URL	three-d.termURL	In a 3-D Secure transaction, this is the URL to where the card

Term	XML Element	Definition
		holder is redirected to after being at the Issuer's page.
Transaction		An act between a [Merchant] and an [Account Holder] that results in an electronic representation of the [Account Holder]'s promise to pay for goods or services received from the act. This includes all [Transaction Type]s, including Purchases, Refunds, and Chargebacks. Each Transaction is issued a unique [Transaction Identifier]. If there are two steps required for funds to transfer, such as a pre-authorization followed by a capture, these are considered 2 transactions.
Requested Amount	requested-amount	The only amount that accompanies the [Transaction] when it is created and/or requested. In the case of a Sale or Refund, this is what the [Merchant] requests. In the case of a [Chargeback], this is the amount that is being contested.
Transaction State	state	The current status of a [Transaction]. Typically, a [Transaction] will start from an "In-Progress" state, and then finish in either the "Success" or "Failed" state.
Status Code	statuses.status-code	The status of a [Transaction]. This is primarily used in conjunction with the [Transaction State] to determine the exact details of the status of the [Transaction].
Success Redirect URL	payment.success-redirect-url	The URL to which the [Account Holder] will be re-directed after a successful payment. This is normally a success confirmation page on the [Merchant]'s website.
Status Description	statuses.status-description	Text used to describe the [Transaction Status].
Status Severity	statuses.status-severity	The severity of the [Transaction], can be information, warning, error.
Transaction Type	transaction-type	The type of [Transaction] that determines its behaviour in [Transaction Processing] and [Merchant Settlement].Examples are: [Authorization], [Capture], [Credit], [Purchase], [Refund], [Void Authorization], [Void Capture], [Void Credit], [Void Purchase].
Xid	three-d.xid	In a 3-D Secure process, this is the unique transaction identifier.

18 Appendix I: iDEAL Participating Banks

BIC	Name
ABNANL2A	ABN Amro Bank
ASNBNL21	ASN Bank
FRBKNL2L	Friesland Bank
INGBNL2A	ING
RABONL2U	Rabobank
SNSBNL2A	SNS Bank
RGGINL21	Regio Bank
TRIONL2U	Triodos Bank
FVLBNL22	Van Lanschot Bankiers
KNABNL2H	Knab

19 Appendix J: List of SEPA Direct Debit and Credit Transfer Return Codes

EE Status Code	Reason Name	Reason Description
201.0001	Return reason not specified	Reason for the return was not specified by the debtor bank
201.0002	Debtor / Account holder deceased	Debtor or account holder is deceased
201.0003	Debtor bank details incorrect	Debtor bank details are incorrect
201.0004	Debtor account closed	Debtor account is closed
201.0005	Insufficient funds on debtor account	There are insufficient funds on the debtor bank account
201.0006	Mandate not valid, not active or cancelled	Mandate reference provided is not valid, not active or cancelled
201.0007	Blocked account	The account has been blocked or frozen
201.0008	Transaction forbidden	The payment type is not allowed for this type of account
201.0009	Invalid Bank Operation Code	Invalid transaction code or incorrect data format (issued by debtor bank)
201.0010	Duplicate transaction on debtor account	A duplicate transaction has been found on the debtor account
201.0011	creditor address missing	Missing creditor address (only in the case of direct debit)
201.0012	missing mandatory mandate information	Incomplete or incorrect mandatory information on the mandate (issued by debtor bank)
201.0013	Invalid File Format	Data format is invalid for any reason other than grouping indicator
201.0014	Refund request by End-Customer (payer)	A refund has been requested by the debtor (payer)
201.0015	Regulatory reasons	Refusal due to regulatory reasons
201.0016	Invalid due date or execution date	The due date or execution date specified in the request is not within the limits required by the payment method

201.0017	Incorrect BIC	Bank Identifier Code (BIC) is incorrect or invalid
201.0018	Amendment of mandate reference	There has been an amendment to the mandate reference provided
201.0019	Returned due to technical problems	The transaction has been returned due to technical problems.
201.0020	Fraudulent Payment	The transaction has been returned due to expected fraud
201.0021	Incorrect Agent	Incorrectly imployed agent
201.0022	Incorrect Currency	The incorrect currency was used
201.0023	Request to cancel	Request to cancel due to investigation
201.0024	Undue Payment	Undue Payment
201.0025	Unrecognized Initiating Party	The initiating party cannot be recognized
201.0026	Invalid debtor account type	Unable to recognize the debtor. The payer is a consumer.
201.0027	Invalid Local Instrument Code	Incorrect Direct Debit type. COR1 used when no COR1 agreement in place.