



Payment Processing API

Merchant Specification

Version: 1.23

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.

Copyright © 2016 Wirecard AG All rights reserved.

Printed in Germany / European Union

Version 1.23

Last Updated: February 25, 2016

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.....	9
1.1	Audience.....	9
1.2	Revision History.....	9
2	Integration Options	10
2.1	Overview.....	10
2.2	Payment Page.....	10
2.3	REST API.....	11
2.4	Batch Processing API	11
2.5	SDK.....	11
3	Payment Page.....	12
3.1	Hosted Payment Page	12
3.2	Integration of Hosted Payment Page (HPP)	13
3.3	Embedded Payment Page	15
3.4	Seamless integration	18
3.4.1	Workflow of Seamless.....	18
3.4.2	Integration.....	19
3.4.3	Seamless card form integration	19
3.4.4	Decoupling card data collection from payment.....	21
3.4.5	Validation, Language and Custom templates.....	23
3.5	Consumer Redirects	26
3.6	Supported languages and Browsers.....	26
3.6.1	Languages.....	26
3.6.2	Supported browsers – Desktop.....	27
3.6.3	Supported browsers – Mobile.....	27
3.7	Fields.....	27
3.7.1	Request Fields	27
3.7.2	Response Fields	30
3.8	Security.....	32
3.8.1	Secret Key Exchange.....	32
3.8.2	Request Signature	33
3.8.3	Response Signature	33
3.8.4	Signature Generation Code Samples.....	34
3.9	Asymmetric Encryption	35

3.9.1	Security.....	35
3.9.2	Asymmetric Digital Signature.....	35
3.9.3	Asymmetric Key Exchange.....	37
3.10	Payment Selection	37
3.11	Configuration.....	38
3.12	Customizing the Payment Page with CSS	38
3.12.1	Dynamic Custom CSS URL	38
3.12.2	Custom CSS file content restrictions	40
3.12.3	Input Fields for Credit Card	42
3.12.4	Dynamic CSS sample.....	43
4	Payment mobile SDK	45
4.1	Preface.....	45
4.1.1	Supported payment methods and transaction types	45
4.1.2	System Requirements	46
4.2	Payment SDK integraton	46
4.2.1	Calculation of Signature	46
4.3	Android Environment	49
4.3.1	Credit Card Payment	49
4.3.2	Sample for a makeCardPayment method call.....	53
4.3.3	Paypal payment.....	54
4.4	SEPA Direct Debit payment.....	55
4.4.1	Performing SEPA payment	55
4.5	Response Handling	56
4.6	iOS Environment.....	57
4.6.1	Apple Pay.....	57
4.6.2	Credit Card.....	59
4.6.3	Description of the API fields.....	60
4.7	Paypal for iOS	62
4.7.1	Perform Paypal payment.....	62
4.8	SEPA.....	63
4.8.1	Perfom a SEPA payment.....	63
4.9	Test Credentials.....	64
5	Accessing Resources.....	64
5.1	Overview.....	64
5.2	Security.....	65

5.3	Card Security – Tokenization	65
6	Submitting Transactions.....	65
6.1	Overview.....	65
6.2	Create a Payment.....	65
6.2.1	Fields.....	65
6.2.2	Storing Additional Data with Custom Fields.....	72
6.2.3	Instant Payment Notifications.....	73
6.2.4	Sample XML Request and Response – Credit Card	75
6.2.5	Sample NVP Request	76
6.3	Retrieve Transaction by Transaction Id.....	92
6.3.1	Fields.....	92
6.3.2	Sample Request.....	92
6.3.3	Sample Response	92
6.4	Retrieve Transaction by Request Id.....	92
6.4.1	Fields.....	92
6.4.2	Sample Request.....	93
6.4.3	Sample Response	93
6.5	Tokenize Credit Card Information	93
6.5.1	Fields.....	93
6.5.2	Sample XML Request and Response – Tokenize Credit Card	93
6.6	Detokenize Credit Card Information	95
6.6.1	Fields.....	95
6.6.2	Sample XML Request and Response – Detokenize Credit Card	95
7	3-D Secure	96
7.1	Overview.....	96
7.2	Merchant Plug-In.....	97
7.2.1	Wirecard Merchant Plug-In (MPI)	97
7.2.2	3rd Party MPI.....	97
7.3	Process Flow	97
7.4	Check Enrollment	98
7.4.1	Fields.....	98
7.4.2	Sample XML Request and Response	98
7.5	ACS Redirect	99
7.5.1	Fields.....	99
7.5.2	ACS HTTPS Redirect	99

7.5.3	ACS HTTPS Redirect to TERM-URL.....	100
7.6	Check Payer Response.....	100
7.6.1	Fields.....	101
7.6.2	Sample XML Request and Response	101
7.7	Payment Request with PAREs	102
7.8	Payment Request with 3rd Party MPI	103
8	Online Banking Payments.....	104
8.1	Create a Payment for Online Banking Methods.....	104
8.2	Direktüberweisung	105
8.2.1	Fields.....	105
8.2.2	XML- Samples for Direktüberweisung.....	107
8.3	Sofortüberweisung	112
8.3.1	Fields.....	112
8.3.2	XML- Samples for Sofortüberweisung.....	116
8.4	Przelewy24	121
8.4.1	Fields.....	122
8.4.2	XML Samples for Przelewy24 Debit	124
8.4.3	XML- Samples for Przelewy24 Refund Debit.....	127
9	Alternative Schemes Payments	129
9.1	Create a Payment for Wallet Payment Methods	129
9.2	Skrill Wallet.....	130
9.2.1	Fields.....	130
9.2.2	XML- Samples for Skrill Wallet	133
9.3	Skrill Credit	137
9.3.1	Fields.....	137
9.3.2	XML- Samples for Skrill Credit	138
9.4	PayPal	141
9.4.1	Fields for PayPal	141
9.4.2	XML Samples for PayPal	142
9.4.3	Capture, Refund and Void for PayPal.....	146
9.4.4	Reference transactions for PayPal	146
9.4.5	Pre-authorization	149
9.4.6	Cross merchant account references.....	149
9.4.7	Basket items for PayPal	150
9.4.8	Granting 3 rd party permission on PayPal.....	150

9.5	ePlatby.....	153
9.5.1	Fields for ePlatby.....	153
9.5.2	XML Samples for ePlatby.....	154
9.6	Guaranteed Invoice, Direct Debit and Installments	157
9.6.1	Fields for Guaranteed Invoice, Direct Debit and Installments	157
9.6.2	XML Samples for Guaranteed Invoice, Direct Debit and Installments	158
9.6.3	Capture, Refund, Void and Credit for Guaranteed.....	162
9.6.4	Basket items for Guaranteed Invoice, Direct Debit and Installments.....	162
9.7	MasterPass	164
9.7.1	Merchant configuration / Registration Procedure.....	164
9.7.2	Fields for MasterPass	169
9.7.3	Sample XML Request and Response for First initialization	169
9.7.4	Sample XML Request and Response for Subsequent initialization	175
9.8	Carrier Billing	180
9.8.1	Fields for carrier billing.....	180
9.8.2	Debit for Carrier Billing.....	183
9.8.3	XML- Samples for Carrier Billing.....	183
9.8.4	Refund for Carrier Billing.....	184
9.8.5	Refund fields.....	185
9.8.6	XML- Samples for Carrier Billing.....	186
9.9	Trust Pay	191
9.9.1	Fields for Trust Pay	191
9.9.2	XML samples for a debit transaction.....	191
9.9.3	Direct Banking Extension.....	193
9.9.4	Supported areas	196
9.10	Paysafecard.....	198
9.10.1	Fields for Paysafecard.....	199
9.10.2	XML samples.....	199
9.11	3pay	203
9.11.1	Workflow of 3pay	203
9.11.2	Fields.....	204
9.11.3	Sample XML Request and Response for 3pay	207
9.12	Alipay Cross Border.....	214
9.12.1	Fields.....	214
9.12.2	XML Samples for Alipay Cross Border	215

10	Direct Debit and Credit Transfer	219
10.1	Create a Payment for Direct Debit and Credit Transfer	219
10.2	SEPA.....	220
10.2.1	SEPA Direct Debit	220
10.2.2	SEPA Direct Debit Recurring Transactions	230
10.2.3	SEPA Direct Debit Authorization Transactions.....	237
10.2.4	SEPA Credit Transfer.....	244
10.2.5	SEPA Void Functionality	249
10.2.6	Deposits.....	253
10.2.7	Bank Data Validations	254
10.2.8	Cross-Payment-Method Referencing.....	255
10.2.9	Cross-Merchant Referencing.....	260
11	Appendix A: Payment Methods.....	266
12	Appendix B: Transaction Types	268
13	Appendix C: Transaction States.....	275
14	Appendix D: Card Types	275
15	Appendix E: Transaction Statuses	275
16	Appendix F: Countries and Currencies	289
17	Appendix G: 3-D-Secure Process Flow.....	289
18	Appendix H: Glossary.....	290
19	Appendix I: iDEAL Participating Banks.....	293
20	Appendix J: List of SEPA Direct Debit and Credit Transfer Return Codes.....	293
21	Appendix K: Test Access Data and Credentials	295

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
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 <u>Several wording enhancements and reformatting</u>
2015-03-11	1.13	Section 9.4.6 Cross merchant account references added to the PayPal chapter.
2015-03-24	1.13	Chapter 2 Integration Options added. Chapter 3 Payment Page added.
2015-04-10	1.13	Chapter 8.7 MasterPass updated. Appendix B updated with new MasterPass transactions. Chapter 3 Payment Page: integration samples extended.
2015-05-13	1.14	Chapter 8.9 Trust Pay created. Appendix A updated with new TrustPay payment method. Appendix B updated with new TrustPay transactions.
2015-06-10	1.15	Chapter 8.10 Paysafecard created. Appendix A updated with new TrustPay payment method. Appendix B updated with new TrustPay transactions. Appendix E updated with new error message (500.3004).
2015-06-18	1.16	Addition of failed request, response and notification for various types of payment methods.
2015-06-26	1.17	Section 9.2.11: Cross-Merchant Referencing added. Chapter 8.5.4 PayPal updated with new pre-authorization transaction.

2015-08-11	1.18	Chapter 8.8.2 Carrier Billing debit transaction added Chapter 9.2.10: Cross-Payment-Method Referencing added.
2015-09-11	1.19	Appendix K added
2015-09-15		Chapter 8.4.2 extended by Provider Transaction ID information.
2015-09-18		Chapter 8.4.4 Reference Transactions for PayPal added. Supported languages for Payment Page added
2015-11-11	1.20	Mobile SDK added 3pay added Appendix B updated HPP & EPP updated SEPA credit transfer – creditor-id added in fields
2015-11-16	1.21	Section 10.2.1.2 Delayed processing of SEPA direct debit transactions added
2015-12-18	1.22	Section 8.4 Przelewy24 added Minor Layout adjustments Version History truncated Appendix A & B updated
2016-02-24	1.3	iOS mobile SDK added & Android section updated Fields for Skrill Wallet updated Fields for Credit Card SEPA DD/Credit/Deposit Sofort banking updated Payment Page Seamless integration and Dynamic Custom CSS styling Alipay Cross Border 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.

2.2 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 three flavors:

- Hosted Payment Page where buyer is redirected to Wirecard's server to finish the checkout procedure
- Embedded Payment page where payment screen is displayed in overlay directly on merchant's checkout page.

- Seamless integration

Only HTML knowledge is required to integrate.

2.3 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 refers to implementation using REST API, though e.g. description of payment methods is common for both options.

2.4 Batch Processing API

In order to process multiple transactions at the same time, Wirecard's Batch Processing is the ideal solution for both fast and easy to implement offline bulk transaction processing.

Batch processing is possible for the following payment methods:

- SEPA Direct Debit
- SEPA Credit Transfer

Working programmer knowledge is required to integrate this way.

2.5 SDK

Serves as a wrapper for EE payment gateway with PCI V3 compliant native mobile card payment form. SDK is naturally developed for integration into merchant's app as a software library where it covers sending payment requests to payment gateway. It uses REST requests constructed by merchant sent over HTTPS Protocol to Wirecards' Payment Processing Gateway end-point where the REST API is exposed.

Following payment methods are currently supported:

- Credit Card payments
- PayPal
- SEPA Direct Debit
- Apple Pay (in Scope)

Operation system interfaces:

- iOS
- Android

Working programmer knowledge is required to integrate this way.

3 Payment Page

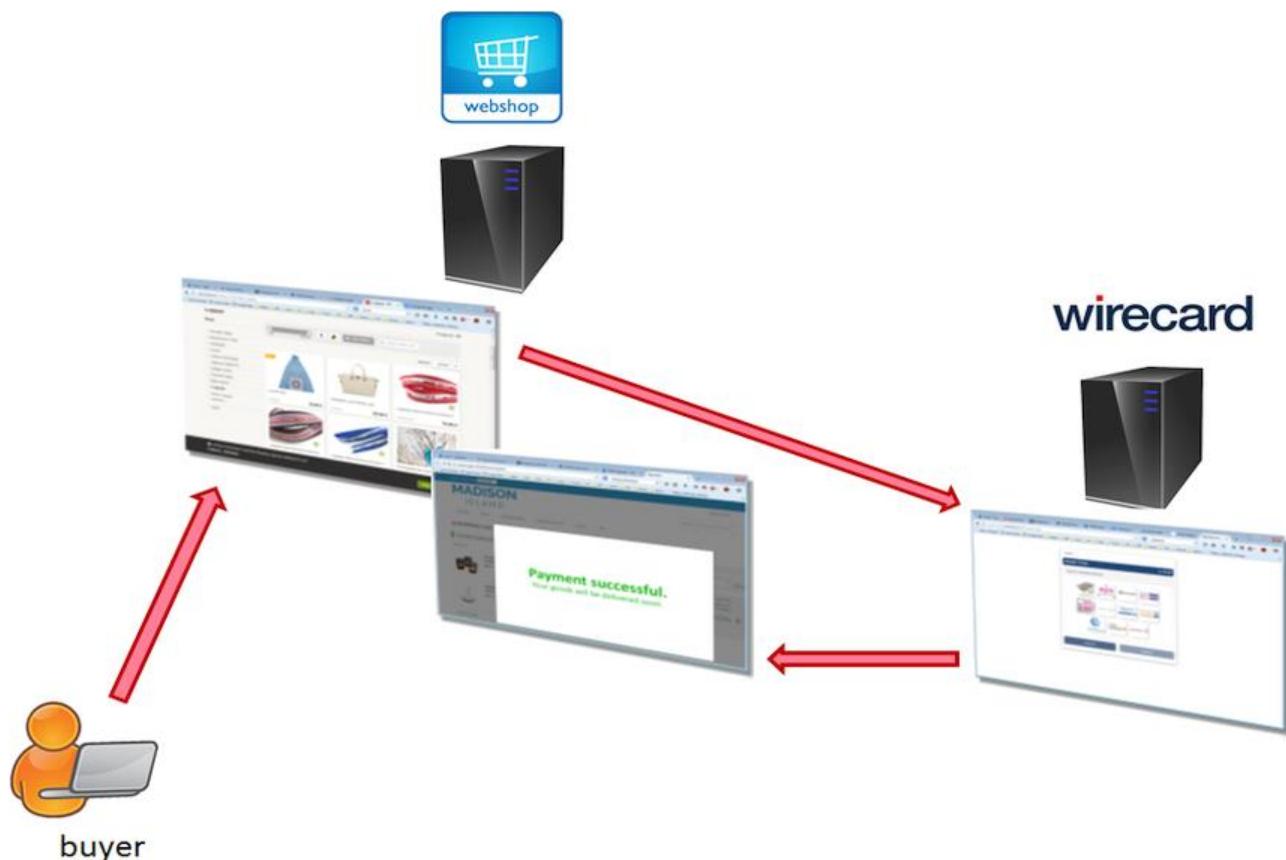
Provides bit more than just an endpoint, it could take care of payment method selection, collecting credit card details or redirecting to alternative payment provider's site. Payment Page comes in three flavors: Hosted Payment Page, Embedded Payment page and Seamless integration.

3.1 Hosted Payment Page

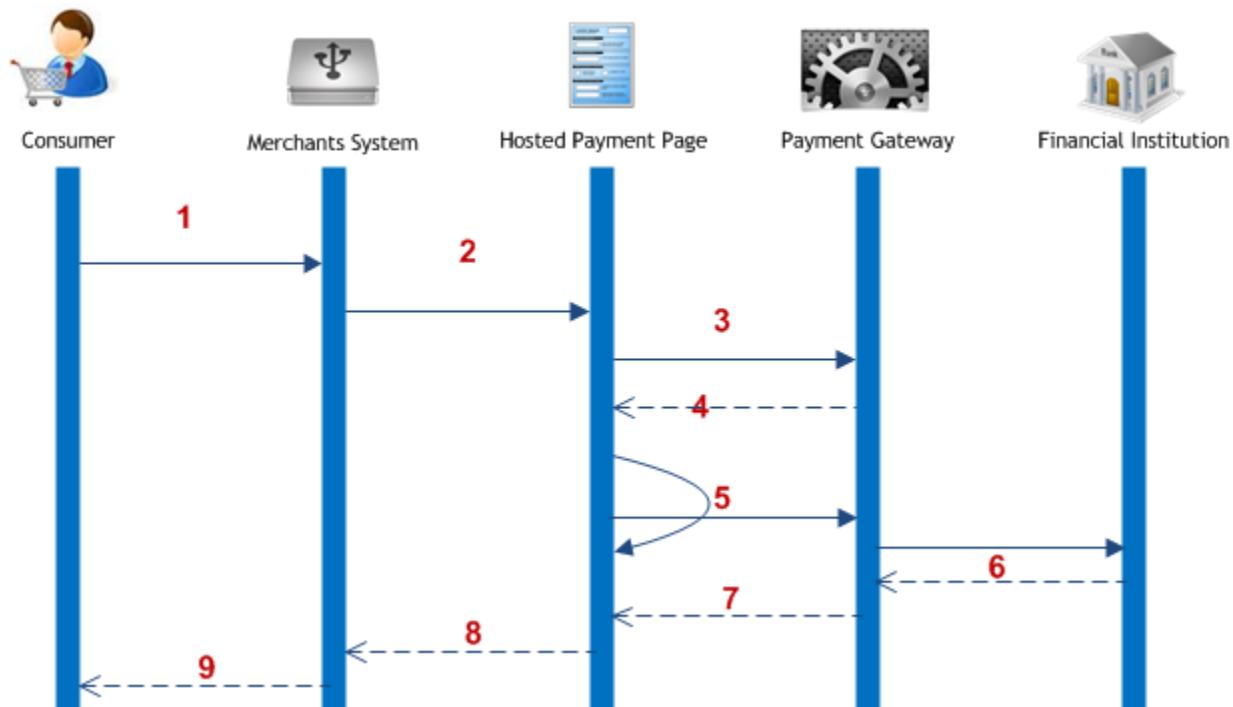
Hosted Payment Page (HPP) integration involves the merchant redirecting the consumer to the Payment Page along with information about the transaction and a secure digital signature.

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/paymentPageLoader.js>

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.

3.2 Integration of Hosted Payment Page (HPP)

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 sample code for redirecting to Hosted Payment Page with payment method selection screen is below.

Integration is as easy as including one JavaScript library into merchant's checkout page,

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

Specification of the payment data

```
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: "e44730486d180cca590bc2e8dea22bd175395636a37b0da0ef785"
}
```

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

```
WirecardPaymentPage.hostedPay(requestData);
```

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).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
    <head>
        <title>
            Demo shop
        </title>
        <script src="https://apitest.wirecard.com/engine/hpp/paymentPageLoader.js" type="text/javascript"></script>
    </head>
    <body>
        <form>
            <input id="wirecard_pay_btn" type="button" onclick="pay()" value="Pay Now"> <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: "purchase",
                        requested_amount: "2.56",
                        requested_amount_currency: "EUR",
                        locale: "en",
                        request_signature:
                        "kg44730486d159df0bc2e8dea22bd175395636a37b0da0ef785"
                    }
                    WirecardPaymentPage.hostedPay(requestData);
                }
            </script>
        </form>
    </body>
</html>
```

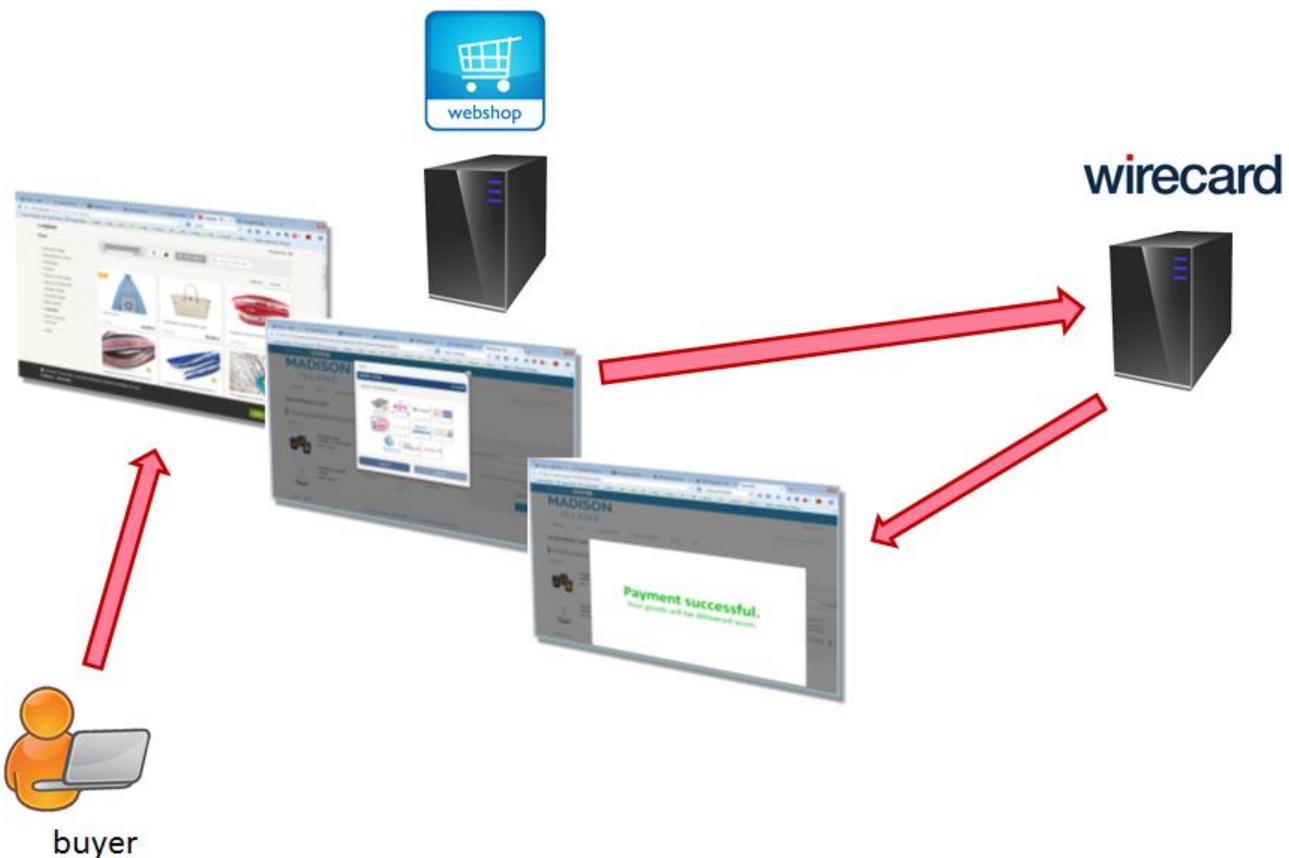
The sample for button redirecting to PayPal checkout portal is below (please note that just fields “payment_method” and “transaction_type” differ).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>
    Demo shop
</title>
<script src="https://api-test.wirecard.com/engine/hpp/paymentPageLoader.js" type="text/javascript"></script>
</head>
<body>
<form>
    <input id="wirecard_pay_btn" type="button" onclick="pay()" value="Pay Now"> <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: "paypal",
                transaction_type: "debit",
                requested_amount: "2.56",
                requested_amount_currency: "EUR",
                locale: "en",
                request_signature:
                "d730486d159df0bc2e8dea22bd175395636a37b0da0ef785"
            }
            WirecardPaymentPage.hostedPay(requestData);
        }
    </script>
</form>
</body>
</html>
```

3.3 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. Embedded Payment Page integration is the same as for Hosted Payment Page – it involves a (different) function call of the same JavaScript library. 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="https://api-test.wirecard.com/engine/hpp/paymentPageLoader.js" type="text/javascript"></script>
```

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

```
WirecardPaymentPage.embeddedPay(requestedData);
```

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>Demo shop</title>
    <script src="https://api-test.wirecard.com/engine/hpp/paymentPageLoader.js" type="text/javascript"></script>
</head>
<body>
    <input id="wirecard_pay_btn" type="button" onclick="pay()" value="Pay Now"/>
    <script type="text/javascript">
        function pay() {
            var requestData = {
                merchant_account_id: "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
                // Other request parameters
            };
        }
    </script>
</body>
</html>
```

```

        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:
        "e44730486d180cca590bc2e8dea22bd175395636a37b0da0ef785"
    }
    WirecardPaymentPage.embeddedPay(requestedData);
}
</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).

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>Demo shop</title>
    <script
        src="https://api-
test.wirecard.com/engine/hpp/paymentPageLoader.js"
        type="text/javascript"></script>
</head>
<body>
    <input id="wirecard_pay_btn" type="button" onclick="pay()" value="Pay Now"/>
    <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: "purchase",
                requested_amount: "2.56",
                requested_amount_currency: "EUR",
                locale: "en",
                request_signature:
                "e44730486d180cca590bc2e8dea22bd175395636a37b0da0ef785"
            }
            WirecardPaymentPage.embeddedPay(requestedData);
        }
    </script>
</body>
</html>

```

3.4 Seamless integration

Seamless integration provides a possibility to incorporate the Wirecard's credit card form into merchant's checkout page for merchants who want to:

- Avoid browser redirects to Wirecard's payment pages and back to the shop during the checkout process.
- Take care of payment selection process and fully control look & feel for card form itself.
- Provide more sophisticated checkout flow that includes collecting card data first, then buyer reviewing the order one more time on a recap page, and only then confirm the payment.
- For credit card payments only

This approach provides a seamless shopping experience for web-shoppers, while the technical solution allows the merchant to fall into a least demanding PCI-DSS category. Main advantages of Seamless solution are then:

- Seamless user experience - card form appears as an integral part of merchant's checkout page.
- Possibility to decouple entering card data from actual payment - allowing to display a summary page in between.
- No redirects in browser, no external processing pages, no pop-up windows.
- Simple Self-Assessment-Questionnaire "A" (SAQ-A) from PCI-DSS is acceptable for merchant certification.

3.4.1 Workflow of Seamless

To tackle the seamless and PCI compliance part, Wirecard serves the complete card form as an HTML iFrame, which ensures that attacker or malicious script won't have access to entered data on merchant's page. Form is then submitted directly to Wirecard's servers over secure TLS channel, so the data in transport are secured. The high-level diagram below shows the checkout flow with order summary page in between the card data collection and actual payment:



1. Customer checks out after shopping spree on merchant side.

2. Merchant's system serves the checkout page, but the complete iFrame with the card form comes from Wirecard server. The form also submits entered data directly to Wirecard server, so it never gets in contact with merchant's system. Only transaction ID and token are returned to merchant.
3. Merchant's system serves order summary page and as soon as buyer confirms the payment, merchant's system makes server call to execute payment either referencing previous zero authorization or using retrieved token.
4. Merchant's system serves success page.

3.4.2 Integration

Integration is done via JavaScript library:

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

There are three functions provided by the library:

WirecardPaymentPage.seamlessRenderForm({ requestData : requestData, wrappingDivId : "seamless-target", onSuccess : processSucceededResult, onError : processErrorResult });	Renders the form. Parameters are: - request data, same as for Hosted Payment Page and similar to REST API integration - id of the element where the form will be rendered - functions to handle call-back data (response is the only parameter of the functions) in cases of success and error
WirecardPaymentPage.seamlessSubmitForm({ requestData : requestData, onSuccess : processSucceededResult, onError : processErrorResult });	Submits the form. Parameters are: - additional request data, only optional - functions to handle call-back data (response is the only parameter of the functions) in cases of success and error
WirecardPaymentPage.seamlessPay({ requestData : requestData, onSuccess : processSucceededResult, onError : processErrorResult });	Submits the payment request. Parameters are: - request data, same as for Hosted Payment Page and similar to REST API integration - functions to handle call-back data (response is the only parameter of the functions) in cases of success and error

3.4.3 Seamless card form integration

This use case covers card form displayed directly on merchant's checkout page for seamless user experience (no browser redirects).

3.4.3.1 Render Form

Merchant renders the form into element on the checkout page. The transaction details are specified in `requestData`. The element is identified by `wrappingDivId` parameter.

```
var requestData = {
```

```

"request_id" : "217c3832-8575-c1d5-0e3a-2fa08003b0fd",
"request_time_stamp" : "20151103095835",
"merchant_account_id" : "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
"transaction_type" : "purchase",
"requested_amount" : "2.00",
"requested_amount_currency" : "EUR",
"ip_address" : "127.0.0.1",
"request_signature" :
"43fd8c261a8fdad693024c7cc009239ad344f1e5716de0c3163237b5392d5700",
"payment_method" : "creditcard"
},
WirecardPaymentPage.seamlessRenderForm({
  requestData : requestData,
  wrappingDivId : "seamless-target",
  onSuccess : processSucceededResult,
  onError : processErrorResult
});

```

Sample of transaction data received in the call-back function when the form is successfully rendered:

```

{"ip_address": "127.0.0.1",
"merchant_account_id": "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
"request_id": "217c3832-8575-c1d5-0e3a-2fa08003b0fd",
"requested_amount": "2.00", "requested_amount_currency": "EUR", "status_code_1": "201.0000",
"status_description_1": "The resource was successfully created.", "status_severity_1": "information",
"transaction_id": "5ff1cf52-8471-11e5-95ea-005056b13376",
"transaction_state": "success",
"transaction_type": "check-signature"}

```

3.4.3.2 Submit form

Merchant assigns the function call to a pay button on the checkout page. `requestData` value is optional for submitting form. It will be merged with data from the first step. Form input is validated before submit.

```

WirecardPaymentPage.seamlessSubmitForm({
  onSuccess : processSucceededResult,
  onError : processErrorResult
});

```

Sample of transaction data received in the call-back function when the payment is successfully submitted

```

{"api_id": "elastic-api",
"authorization_code": "153620", "completion_time_stamp": "20151106103540", "first_name": "John",
"ip_address": "127.0.0.1", "last_name": "Doe", "masked_account_number": "444433*****1111", "merchant_account_id": "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
"payment_method": "creditcard", "provider_transaction_id_1": "8ee6e543-c96f-459a-a379-50d0e9152fd3", "request_id": "217c3832-8575-c1d5-0e3a-2fa08003b0fd",
"requested_amount": "2.00", "requested_amount_currency": "EUR", "self": "https://test-api.wirecard.com:9000/engine/rest/merchants/61e8c484-dbb3-4b69-ad8f-706f13ca141b/payments/1fe8a33a-8472-11e5-95ea-005056b13376", "status_code_1": "201.0000",
"status_message": "The payment has been successfully submitted."}

```

```
"status_description_1": "3d-acquirer:The resource was successfully created.", "status_severity_1": "information", "token_id": "4186409015611111", "transaction_id": "1fe8a33a-8472-11e5-95ea-005056b13376", "transaction_state": "success", "transaction_type": "purchase"}
```

3.4.4 Decoupling card data collection from payment

This use case covers merchant displaying a recap order page in-between card data collection from buyer and actually submitting the payment. In order to do so, merchant have an option to use card form to perform:

- Zero authorization transaction that could be later referenced by a regular authorization or purchase. Advantage is that zero authorization transaction is confirmed by an issuer, and CVC is checked in this case.
- Tokenization only transaction. In that case a token (non-sensitive data) is returned to merchant who can use it for a real authorization or purchase later.

In either case, sensitive card data will not touch any of the merchant's systems.

The difference to previous use case is that “authorization-only” transaction with zero amount is used instead of “purchase” (“tokenization” transaction could be used to a similar effect), and the real fund booking only comes in the third (additional) step. Card form is displayed directly on merchant’s checkout page.

3.4.4.1 Render form

Merchant renders the form into element on their checkout page. The transaction details are specified in `requestData`. The element is identified by `wrappingDivId` parameter.

```
var requestData = {
  "request_id": "217c3832-8575-c1d5-0e3a-2fa08003b0fd",
  "request_time_stamp": "20151103095835",
  "merchant_account_id": "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
  "transaction_type": "authorization-only",
  "requested_amount": "0",
  "requested_amount_currency": "EUR",
  "ip_address": "127.0.0.1",
  "request_signature": "",
  "payment_method": "creditcard"
};

WirecardPaymentPage.seamlessRenderForm({
  requestData: requestData,
  wrappingDivId: "seamless-target",
  onSuccess: processSucceededResult,
  onError: processErrorResult
});
```

Sample of transaction data received in the call-back function when the form is successfully rendered:

```
{"ip_address": "127.0.0.1", "merchant_account_id": "61e8c484-dbb3-4b69-ad8f-706f13ca141b", "request_id": "217c3832-8575-c1d5-0e3a-2fa08003b0fd", "requested_amount": "0", "requested_amount_currency": "EUR", "status_code_1": "201.0000", "status_description_1": "The resource was successfully created.", "status_severity_1": "information", "transaction_id": "5ff1cf52-8471-11e5-95ea-005056b13376", "transaction_state": "success",}
```

```
"transaction_type": "check-signature"}
```

3.4.4.2 Submit form

Merchant assigns the function to a pay button on the checkout page. `requestData` value is optional for submitting form. It will be merged with data from the first step. Form input is validated before submit.

```
WirecardPaymentPage.seamlessSubmitForm({
  onSuccess : processSucceededResult,
  onError : processErrorResult
});
```

Sample of transaction data received in the call-back function when the payment is successfully submitted

```
{"api_id": "elastic-api",
"authorization_code": "153620",
"completion_time_stamp": "20151106103540",
"first_name": "John", "ip_address":
"127.0.0.1", "last_name": "Doe",
"masked_account_number": "444433*****1111",
"merchant_account_id":
"61e8c484-dbb3-4b69-ad8f-706f13ca141b", "payment_method":
"creditcard", "provider_transaction_id_1":
"8ee6e543-c96f-459a-a379-50d0e9152fd3", "request_id":
"217c3832-8575-c1d5-0e3a-2fa08003b0fd", "requested_amount":
"0", "requested_amount_currency": "EUR",
"self":
https://test-api.wirecard.com:9000/engine/rest/merchants/61e8c484-dbb3-4b69-ad8f-706f13ca141b/payments/1fe8a33a-8472-11e5-95ea-005056b13376,
"status_code_1": "201.0000",
"status_description_1": "3d-acquirer:The resource was
successfully created.", "status_severity_1":
"information", "token_id": "4186409015611111",
"transaction_id": "1fe8a33a-8472-11e5-95ea-005056b13376",
"transaction_state": "success",
"transaction_type": "authorization-only"}
```

3.4.4.3 Actual payment

Merchant can now reference existing authorization-only transaction to do a payment without buyer's further interaction. Merchant can do that also using function of Wirecard's JavaScript library from the browser.

```
var requestData = {
  "request_id" : "679047dc-8a4d-657b-91dc-df6d80cd6a10",
  "request_time_stamp" : "20151103101310",
  "merchant_account_id" : "61e8c484-dbb3-4b69-ad8f-706f13ca141b",
  "transaction_type" : "purchase",
  "requested_amount" : "12",
  "requested_amount_currency" : "EUR",
  "ip_address" : "127.0.0.1",
  "request_signature" :
"ea468004287bf191a43b4c5ac33e38d5f035050ad784cad293646c1533e7fc48",
  "payment_method" : "creditcard",
  "parent_transaction_id" : "26c14051-8213-11e5-a96e-0050b667eb91"
};
WirecardPaymentPage.seamlessPay({
  requestData : requestData,
  onSuccess : processSucceededResult,
```

```
    onError : processErrorResult
});
```

Sample of transaction data received in the call-back function when the payment is successfully submitted

```
{"api_id": "elastic-api",
"authorization_code": "153620",
"completion_time_stamp": "20151106103540",
"first_name": "John", "ip_address":
"127.0.0.1", "last_name": "Doe",
"masked_account_number": "444433*****1111",
"merchant_account_id":
"61e8c484-dbb3-4b69-ad8f-706f13ca141b", "payment_method":
"creditcard", "provider_transaction_id_1":
"8ee6e543-c96f-459a-a379-50d0e9152fd3", "request_id":
"217c3832-8575-c1d5-0e3a-2fa08003b0fd", "requested_amount":
"12", "requested_amount_currency": "EUR",
"self":
https://test-api.wirecard.com:9000/engine/rest/merchants/61e8c484-dbb3-4b69-ad8f-706f13ca141b/payments/1fe8a33a-8472-11e5-95ea-005056b13376,
"status_code_1": "201.0000",
"status_description_1": "3d-acquirer:The resource was
successfully created.", "status_severity_1":
"information", "token_id": "4186409015611111",
"transaction_id": "1fe8a33a-8472-11e5-95ea-005056b13376",
"transaction_state": "success",
"transaction_type": "purchase"}
```

3.4.5 Validation, Language and Custom templates

Form Validation

Merchant can request form validity by using function.	<pre>WirecardPaymentPage.seamlessFormIsValid({ onValidationResult : processValidationResult, })</pre>
Merchant can validate the form input anytime calling function. The only parameter is the function to handle call-back data.	<pre>WirecardPaymentPage.seamlessValidateForm({ onValidationResult : processValidationResult, })</pre>

Language

Merchant can change form's locale using function. The only parameter is language ISO code.	<pre>WirecardPaymentPage.seamlessChangeLocale("en");</pre>
--	--

Custom templates

MERCHANTS ARE ABLE TO SPECIFY THEIR OWN CUSTOM TEMPLATES TO BE USED INSTEAD OF DEFAULT WIRECARD'S CARD FORM. TEMPLATE CONSISTS OF HTML CONTENT, CSS AND JAVASCRIPT PARTS. THE FOLLOWING TEMPLATE IS AN EXAMPLE OF CUSTOM MERCHANT'S TEMPLATE THAT USES BOOTSTRAP AND FORM VALIDATION LIBRARIES. FIRST, HTML PART WILL BE INSERTED INTO <BODY> TAG OF THE PAGE, SO IT IS A SIMPLE HTML FORM:

```
<h3 data-i18n="cc_form_title">Billing information</h3>
<form id="seamless-form" data-wd-validate-form="true" th:object="${payment}">
    <div class="row">
        <div class="form-group col-xs-12">
            <div class="card-types btn-group" data-toggle="buttons">
                <label class="btn btn-default active"> <input type="radio" class="ee-request-nvp" name="card_type" value="visa" checked="checked" data-fv-wdcardtype="true" data-fv-wdcardtype-message="select_valid_card_type" data-fv-wdcardtype-creditcardfield="account_number" /> 
                </label> <label class="btn btn-default"> <input type="radio" class="ee-request-nvp" name="card_type" value="mastercard" /> 
                </label> <label class="btn btn-default"> <input type="radio" class="ee-request-nvp" name="card_type" value="diners" /> 
            </label>
            </div>
        </div>
        <div class="row">
            <div class="col-sm-8">
                <div class="form-group">
                    <input type="text" class="form-control ee-request-nvp" id="account_number" name="account_number" placeholder="Card number" data-i18n="card_number" data-fv-wdcreditcard="true" data-fv-wdcreditcard-message="enter_creditcard_number" data-fv-wdcreditcard-cardtypefield="card_type" data-fv-wdcreditcard-cvvfield="card_security_code" data-fv-wdcreditcard-allowedcardtypes="visa,mastercard,diners" data-fv-onsuccess="FormValidation.WDHelper.onCardNumberSuccess" data-fv-onerror="FormValidation.WDHelper.onCardNumberError" />
                </div>
            </div>
            <div class="col-sm-4">
                <div class="form-group">
                    <input type="text" class="form-control ee-request-nvp" id="card_security_code" name="card_security_code" placeholder="CVV" data-fv-wdcvv="true" data-fv-wdcvv-message="enter_cvv" data-fv-wdcvv-creditcardfield="account_number" data-i18n="card_security_code" />
                </div>
            </div>
        </div>
        <div id="expiry-date-div" class="row">
            <div class="col-sm-8"></div>
            <div class="col-sm-2">
                <div class="form-group">
                    <input type="number" class="form-control ee-request-nvp" id="expiration_month" name="expiration_month" placeholder="Month" data-i18n="expiration_month" data-fv-notempty="true" data-fv-notempty-message="enter_value" data-fv-wdexpirymonth="true" />
                </div>
            </div>
        </div>
    </div>
</form>
```

```

        data-fv-wdexpirymonth-message="invalid_expiry_date"
        data-fv-wdexpirymonth-yearfield="expiration_year" />
    </div>
</div>
<div class="col-sm-2">
    <div class="form-group">
        <input type="number" class="form-control ee-request-nvp"
            id="expiration_year" data-fv-notempty="true"
            data-fv-notempty-message="enter_value" data-fv-wdexpiryyear="true"
            data-fv-wdexpiryyear-message="invalid_expiry_date"
            data-fv-wdexpiryyear-yearscount="20"
            data-fv-wdexpiryyear-monthfield="expiration_month"
            name="expiration_year" placeholder="Year"
            data-i18n="expiration_year" />
    </div>
</div>
</div>
<div class="row">
    <div class="col-sm-6">
        <div class="form-group">
            <input type="text" class="form-control ee-request-nvp"
                id="first_name" name="first_name"
                placeholder="Cardholder first name" data-i18n="first_name"
                th:value="*{accountHolder} ? *{accountHolder.firstName}" />
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group">
            <input type="text" class="form-control ee-request-nvp"
                id="last_name" name="last_name" placeholder="Cardholder last name"
                data-fv-notempty="true" data-fv-notempty-message="enter_value"
                data-i18n="last_name"
                th:value="*{accountHolder} ? *{accountHolder.lastName}" />
        </div>
    </div>
</div>
</form>

```

The HTML part may be completed with CSS file:

```

body {
    font-family: Verdana;
    font-size: 12px;
    padding: 20px;
}
input {
    width: 100%;
}
.btn {
    outline: none !important;
}
.card-types .btn {
    height: 40px
}
.card-types .btn img {
    width: 40px;
}
.card-types .form-control-feedback {
    right: -40px !important;
    top: 1px !important;
}

```

Each merchant could provide multiple templates and specify which one to use when rendering Seamless form. Please contact Customer support in order to set up custom templates.

3.5 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 [3.8.3 Response Signature](#)). It is strongly recommended for merchants to 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.

3.6 Supported languages and Browsers

3.6.1 Languages

Based on the end-consumers browser language settings both Hosted Payment Page and Embedded Payment Page set the appropriate language automatically. The Following table shows the supported languages.

Code	English Name	Local Name
ar	Arabic	عربی
de	German	Deutsch
en	English	English
es	Spanish	Español
fr	French	Français
it	Italian	Italiano
iw	Hebrew	עברית
ja	Japanese	日本語
ka	Georgian	საქართველოს
nl	Dutch	Nederlands
pl	Polish	Polski
ru	Russian	Русский
uk	Ukrainian	Українська
zh_CN	Chinese (Simplified)	中文 (简体)
zh_TW	Chinese (Traditional)	中文 (繁体)

3.6.2 Supported browsers – Desktop

Browser name	Browser Versions	OS Platform	OS Version	Supported
Chrome	33.0 - 44.0	Windows	7 / 8 / 8.1	✖
Firefox	28.0 - 39.0		7 / 8 / 8.1	✖
Internet Explorer	7 - 11		7 / 8 / 8.1	✖
Internet Explorer	7		XP	✗
Opera	12.15 - 12.17		7 / 8.1	✖
Edge	12		10	✖
Safari	5.1 - 8	OS X	Snow Leopard - Yosemite	✖

3.6.3 Supported browsers – Mobile

OS Platform	OS Version	Supported
Android	4 – 4.4	✖
MAC	5.1. – 8.3.	✖

3.7 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.

3.7.1 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	Please see: 3.8.2 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. Merchants 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.

3.7.2 Response Fields

Field	Cardinality	Datatype	Size	Description
-------	-------------	----------	------	-------------

response_signature	Mandatory	Alphanumeric	64	Please see: 3.8.3 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	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.

3.8 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.

There are two types of signatures

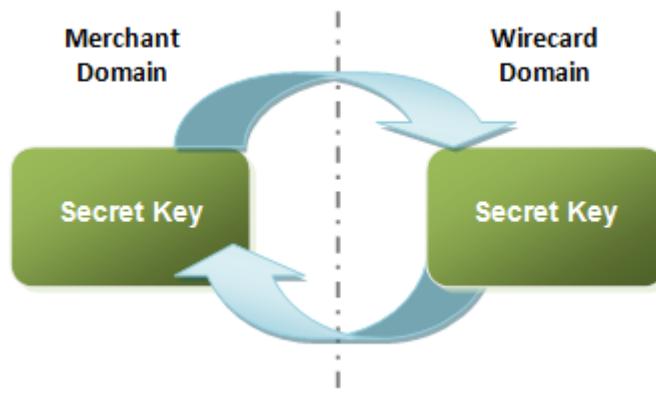
- Request Signature
- Response Signature

3.8.1 Secret Key Exchange

To ensure the authenticity of the request and response messages, it is required that a Secret Key is shared with the merchant.

The Secret Key is used in generation of the `request_signature` and `response_signature` fields.

It is important that the Secret Key is never shared with anyone and is protected within the Merchant Website (only used in server side code for generating the request signature or validating the response signature).



The Secret Key will be communicated at the time of Merchant Account setup. Please contact your support representative if you did not receive a Secret Key or require Secret Key regeneration.

3.8.2 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
```

3.8.3 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)

3.8.4 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

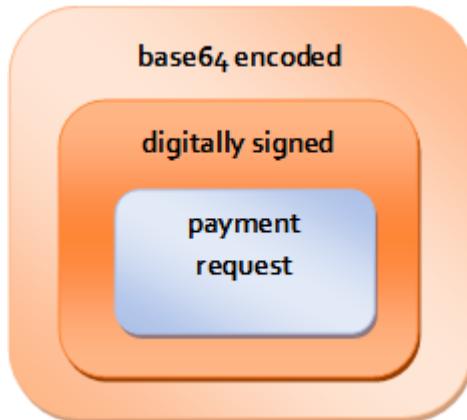
```
import java.security.MessageDigest;
...
def messageDigest = MessageDigest.getInstance("SHA256");
def secret_key = 'XXXXXXXXXXXXXXXXXXXXXX';
def stringToHash = time_stamp + request_id + merchant_account_id + transaction_type
+ requested_amount + requested_amount_currency + redirect_url + ip_address +
secret_key;
messageDigest.update( stringToHash.trim().getBytes() );
def shaHex = new BigInteger(1, messageDigest.digest()).toString(16);
```

3.9 Asymmetric Encryption

3.9.1 Security

Transmitted messages must have the following attributes:

- Payment request message as described here.
- Digitally signed with either asymmetric (DSA or RSA)
- Base64 encoded

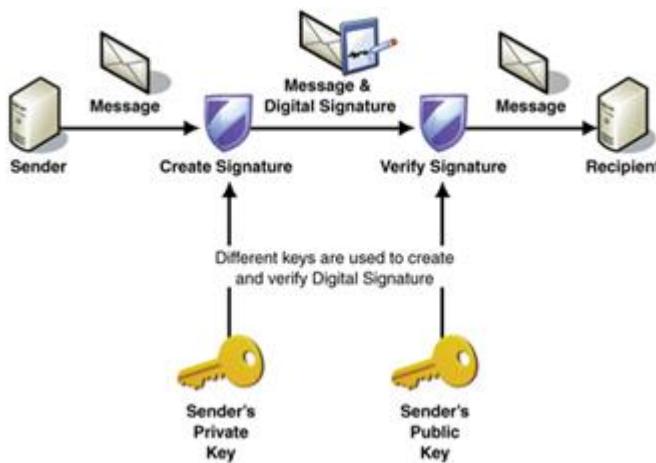


3.9.2 Asymmetric Digital Signature

The Payment Pages use an XML signature for all message exchange between systems. Both DSA and RSA keys are supported.

An XML signature is enclosed in the XML envelop of the message payload. RFC 2828 defines a digital signature as "a value computed with a cryptographic algorithm and appended to a data object in such a way that any recipient of the data can use the signature to verify the data's origin and integrity".

The high level process of asymmetric signature generation and validation is depicted below:



The XML signature has advantages over any proprietary signing/hashing algorithm. Open standard libraries are available that make signature generation and the validation process rather seamless. Secondly, the signature contains meta-information about the cryptographic algorithms used and which references are used for the signature generation and validation. This eliminates the need of prior negotiation for the signature strategies.

XML signatures are often described as being of one or more of three types:

- A detached signature is over data that is external to the Signature element. This could be data outside of the document.
- An enveloping signature is over data that is inside the Signature element.
- An enveloped signature is a signature that is over data that contains the Signature element itself, such as the entire document.
- An enveloped signature is used for the Hosted Payment Page message exchange.

Sample

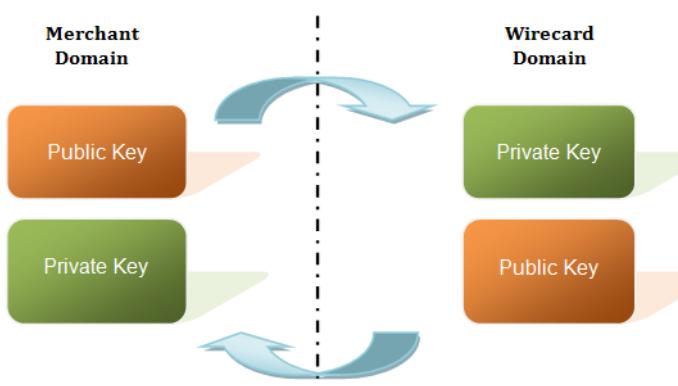
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>c3671cf9-c775-4e39-8d67-31ce24094682</merchant-account-id>
  <request-id>c402a2b3-e17f-468b-ab4f-a84059f03120</request-id>
  <transaction-type>purchase</transaction-type>
  <requested-amount currency="EUR">0.50</requested-amount>
  <account-holder> ...
  </account-holder>
  <ip-address>127.0.0.1</ip-address>
  <order-number>PO98U457Z</order-number>
  <order-detail>Mission Impossible Box DVD Set</order-detail>
  <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#dsa-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"/>
      </Reference>
    </SignedInfo>
    <SignatureValue>...
    </SignatureValue>
  </Signature>
</payment>
```

```

<DigestValue>w7/Dbc3fIlyY21vGT7Sz7GBD1cA=</DigestValue>
</Reference>
<SignedInfo>
<SignatureValue>Jrgj29KYcmgKmgwOCzeOQt1Evw4S3XrL+dPP3ffTh4+IFHpuBo0mSA==</SignatureValue>
<KeyInfo>
<X509Data>
<X509SubjectName>CN=John Doe,OU=Technology,O=Merchant,L=Location,ST=ON,C=CA </X509SubjectName>
<X509Certificate>MIIC0zCCAfmgaWIBAgIETPrqEDALBgcqhkjOOAQBBQAwZTE
LMAkGA1UEBhMCQ0ExCzAJBgNVBA
gTAK9OMRAwDgYDVQQHEwdUb3JvbnnRvMQ0wCwYDVQQKEwRXUFNMMRMwEQYDVQQLEw
pUZWNobm9sb2d5MRMwEQYDVQQDEwp
wpNYW5vaiBTYWh1MB4XDTEwMTIwNTAxMjUzNl0XDTExMTEzMDAxMjUzNl0wZTEL
AkGA1UEBhMCQ0ExCzAJBgNVBAgT
Ak9OMRAwDgYDVQQHEwdUb3JvbnnRvMQ0wCwYDVQQKEwRXUFNMMRMwEQYDVQQLEwpU
ZWNobm9sb2d5MRMwEQYDVQQDEwp
NYW5vaiBTYWh1MIHwMIGoBgcqhkjOOAQBMIGcAkEA/KaCzo4Syrom78z3EQ5SbbB
4sF7ey80etKII864WF64B81uRpH
5t9jQTxeEu0ImbzRMqzVDZkVG9xD7nN1kuFwIVAJYu3cw2nLqOuyYO5rahJtk0bj
jFAkBnhHGyepz0TukaScUUfbGpq
vJE8FpDTWSGkx0tFCcbnjUDC3H9c9oXkGmzLik1Yw4cIGI1TQ2iCmxBblC+eUykA
OMAAkAND242NWsmu/XvvfipM8W2
BSxjP5Bd1I74n7PvLV/ELaDCj0jZztsHneJGiLqww8pDKbz8du8zOJ0J6pQSGS0k
MAsGBYqGSM44BAEFAAMvADAsAhQ
teA+mc04z7tDkNFi6dpopW07bkAIUVzbDKXqudXjuqr/hXPDk1o5xVE4=</X509C
ertificate>
</X509Data>
</KeyInfo>
</Signature>
</payment>
```

3.9.3 Asymmetric Key Exchange

Since the public-key infrastructure would be used for all message exchange, it is required that public keys are exchanged between the merchant and Wirecard's Payment Processing Gateway. This can be done either through email or an URL where the certificate can be downloaded.



3.10 Payment Selection

Payment Page displays payment method 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 overridden 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.

3.11 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.

3.12 Customizing the Payment Page with CSS

CUSTOM CSS URL feature provides the option to send external CSS file link. Because of this feature **Merchant is able to customize his own Payment Page** by editing existing tags styles from external css file. **Merchant is also able to unhide optional field** on Credit Card payment page. And there is an **option to set up custom logo** on Payment Process Page.

Advantage of this functionality is to send different types of CSS for different kind of requests. Second benefit is to customize your theme without asking us for permission. Your developer just need to hold a few rules and restrictions, which are not in fact any kind of obstacles to get your page designed into final deserved looking THEME.

Hosted and Embedded Payment Page is set up to read default css file **embeddedHpp.css**. This file handles basic default Color schemas. Developers must be obvious, because few of basic structures are included in bootstrap files.

Main <form> container is based on **BOOTSTRAP Grid systems**, which are used for creating page layouts through a series of rows and columns that house Payment Page content.

Developers are able to customize **bootstrap.min.css**, so during customizing theme, developer would need to observe bootstrap grid system usage rules. For more detail about bootstrap CSS conditions and usage please visit bootstrap page on <http://getbootstrap.com/css/>

3.12.1 Dynamic Custom CSS URL

Dynamic Custom CSS URL is now Hidden Mandatory Field named as “custom_css_url”, which is also part of the Signature. Dynamic CSS URL must meet the following criteria.

This functionality has been implemented to overlap default styling definitions in embeddedHpp.css. and bootstrap.min.css. Custom css would appear on the bottom of css files in head tag. This fact guarantee casting existed CSS declaration.

1. Custom CSS URL must be shorter than 256 characters including non-alfa-numerical characters. In other cases developer would be notified by error message: Custom CSS URL is too long - more than 256 characters.

2. In general URIs as defined by RFC 3986 may contain any of the following characters: A-Z a-z 0-9 -._~:/?#[]@!\$&'()*+,;=.

In other cases developer would be notified by error message: Custom CSS URL has invalid format.

3. Custom CSS file must be smaller than 50 kb. In other cases developer would be notified by error message: Custom CSS is too large. Max. 50 kB

4. Custom URL needs to be send without any mistakes, or more precisely URL must be in valid format. In other cases developer would be notified by error message: Custom CSS cannot be fetched

5. Custom URL must be stored on server, which is running on valid SSL certificated. Custom CSS URL must be requested over HTTPS protocol. In other cases developer would be notified by error message: Custom CSS URL must be loaded from HTTPS source with valid SSL certificate.

6. Merchant is unable to request psp name and custom css url at the same time. If both are requested only custom css url will be accepted.

7. Merchant needs to create a digital signature on his own server as it is the only place where the Secret is stored. The client's server also provides the Merchant Account ID to the mobile app client.

To calculate the signature, following conditions need to be met:

- Fields need to be concatenated,
- leading and trailing space removed,
- SHA-256 signature follows the combined string

Please note that 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).

Field order

1. request_time_stamp
2. request_id
3. merchant_account_id
4. transaction_type
5. requested_amount
6. requested_amount_currency
7. redirect_url (optional)
8. custom_css_url (optional)
9. ip_address (optional)
10. [secretkey]

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                = 'https://test.com'
custom_css_url               = 'https://test.com/custom.css'
ip_address                  = '127.0.0.1'
secret_key                  = 'efabf47b-e43b-4785-873f-1c5bc65b7cd2'

Pre SHA-256 string
20120430123012order-12345b19fb056-d8da-449b-ac85-
cfbfd0558914purchase1.01USDhttps://test.comhttps://test.com/custom.css127.0.0.1e
fabf47b-e43b-4785-873f-1c5bc65b7cd2

SHA-256 signature
a186cd295f5b0da14aa158090ee8abfcc1ca22961f2c19ea659c4a8f5cbb4a03

```

3.12.2 Custom CSS file content restrictions

1. Custom CSS cannot contains property '-moz-binding'.

CSS, attributes starting with "-" describes a proprietary attribute that might not be supported by all browsers. "-moz-binding" is a attribute used to load another XML file to describe the presentation of the element selected. For example, the following code segment imports an external XML file to apply CSS properties. In case of usage, error message would appear:

Custom CSS contains not allowed property '-moz-binding'.

Invalid usage

```

body {
    -moz-binding: url("external-file.xml#xss");
}

```

2. Custom CSS cannot contains property 'behavior'.

Element 'behavior' is the IE's property with same functionality as -moz-binding. In case of usage, error message would appear:

Custom CSS contains not allowed property 'behavior'.

Invalid usage

```

body {
    behavior:url("script.htc");
}

```

3. For the same security reasons, which is held for properties "-moz-binding" and "behavior", property 'expression' is not allowed to use. . In case of usage, error message would appear:

Custom CSS contains not allowed property value 'expression'.

4. For the same security reasons, which is held for properties "-moz-binding" and "behavior", property ':before' and ':after' are not allowed to use.. In case of usage, error message would appear:

Custom CSS contains not allowed selectors '::after' or '::before'.

5. Some CSS attributes support using "url"s in its declaration. For example, it is able to use a URL when setting the source of an image in CSS. Because of this reason it is prohibited to use 'url(\"javascript:'. property. In case of usage, error message would appear:

Custom CSS contains not allowed property value 'url(\"javascript:'.

6. In CSS, it is option to import another stylesheet into the current stylesheet by prepending the file you want to include in a "@import" declaration at the start of the file. For example, the following CSS includes another CSS file called "included.css". In case of usage, error message would appear:

Custom CSS contains not allowed @import directive.

Invalid usage

```
@import url("included.css");
---existing CSS statements
```

7. It is also unable to use unknown directives. In case of usage, error message would appear:

Custom CSS has been evaluated as harmful because it contains unknown directive.

8. Custom CSS file can't contains unknown properties. In case of usage, error message would appear:

Custom CSS cannot be parsed.

9. All external sources needs to be called from absolute path. Usage of relative paths are not allowed. In case of usage relative path, image will not be loaded at all, and there is no alert message.

10. All pf the external sources need to be loaded from server which is running on valid SSL certificate over HTTPS secured protocol. In case of usage, error message would appear:

Custom CSS URL contains sources with absolute path which are not using HTTPS secure protocol

Invalid usage	Valid usage
<pre>/* ***** MAIN ELEMENTS THEMING - START ***** */ #hpp-logo { /* customizing logo, this field could be hidden */ height: 45px; width: 200px; float: right; background-image: url(your- company-logo.png) !important; /* Caution: relative path is not allowed. To reach proper functionality, image source file has to be called from absolute path over HTTPS secure protocol. */ background-repeat: no-repeat;</pre>	<pre>/* ***** MAIN ELEMENTS THEMING - START ***** */ #hpp-logo { /* customizing logo, this field could be hidden */ height: 45px; width: 200px; float: right; background-image: url(HTTPS://www.your-company- site.com/your-company-logo.png) !important; /* Absolute path has been used over HTTPS protocol */ background-repeat: no-repeat;</pre>

}	}
---	---

3.12.3 Input Fields for Credit Card

The following elements are mandatory/optional for sending a request for the payment method Credit Card. All of the Optional fields are hidden by default. There is an option to unhide Optional fields in Custom CSS by setting div ID parameter display value to block.

Term	Man/Opt	Type	Value	Div tag ID
First Name	O	Input	String	#f_name_id
Last Name	M	Input	String	#l_name_id
Card Type	M	Select	String	<i>„ Explicitly shown „</i>
Card Number	M	Input	Numerical	<i>„ Explicitly shown „</i>
CVV	O	Input	Numerical	#cvv_id
Expiry Date - Month	M	Select	Numerical	<i>„ Explicitly shown „</i>
Expiry Date - Year	M	Select	Numerical	<i>„ Explicitly shown „</i>
Address (1)	O	Input	String	#hpp-creditcard-form-row-for-street1-field
Address (2)	O	Input	String	#hpp-creditcard-form-row-for-street2-field
City	O	Input	String	#hpp-creditcard-form-row-for-city-field
State/Province	O	Input	String	#hpp-creditcard-form-row-for-state-and-postalcode-fields
Postal Code/Zip				
Country	O	Select	String	#hpp-creditcard-form-row-for-country-field
E-mail	O	Input	Valid email address	#hpp-creditcard-form-row-for-email-field
Phone	O	Input	Valid phone number	#hpp-creditcard-form-row-for-phone-field

Example of usage optional email and phone fields for Credit Card Payment Method

```
/* ***** ENABLING OPTIONAL FIELDS - START ***** */
#hpp-creditcard-form-row-for-email-field {
    display : block;
}
#hpp-creditcard-form-row-for-phone-field {
    display : block;
}
/* ***** ENABLING OPTIONAL FIELDS - END ***** */
```

CREDIT CARD

First Name	(optional)	Last Name	
John		Doe	
Card Type			
Visa			
Card Number	CVV	(optional)	
<input type="text" value="Card Number"/>	<input type="text" value="CVV"/>		
Expiry Date	Month	Year	
Address (1)		(optional)	
Mullerstrasse 137			
Address (2)		(optional)	
Address (2)			
City	(optional)		
Berlin			
State/Province	(optional)	Postal Code/Zip	(optional)
Berlin		13353	
Country	(optional)		
Germany			
E-mail	(optional)		
john@doe.com			
Phone	(optional)		
+421123456789			

3.12.4 Dynamic CSS sample

There is prepared offline version of demo page with few comments, how to customize few different tags, to get your deserved theme.

If you're not familiar with few of used properties, please visit <http://www.w3schools.com/css/> for more information.

```
/* ***** MAIN ELEMENTS THEMING - START ***** */
.hpp-logo {
/* customizing logo, this field could be hidden */
height: 45px;
width: 200px;
float: right;
background-image: url(your-company-logo.png) !important; /* Caution: relative path is not allowed. To reach proper
functionality, image source file has to be called from absolute path over HTTPS secure protocol.*/
background-repeat: no-repeat;
background-position: right top;
}
.hpp-template, .hpp-nav > ul.nav > li.active > a {
background: #5FD923 !important;
color: white !important;
```

```
}

.hpp-template {
    background:-webkit-linear-gradient(#333333,#333333)!important;
    background:linear-gradient(#333333,#333333)!important;
    border: 1px solid #333333 !important;
}
.hpp-template-hover:hover, .hpp-template-focus:focus {
    background:#00D469!important;
    background:-webkit-linear-gradient(#00D168, #00964A)!important;
    background:linear-gradient(#00D168, #00964A)!important;
    box-shadow:0px 0px 11px #00994C!important;
}
.hpp-form-cancel{
background:maroon !important;
border: 1px solid maroon !important;
}
.hpp-form-cancel:hover, .hpp-form-cancel:focus {
    background:red important;
    background:-webkit-linear-gradient(red, red)!important;
    background:linear-gradient(red, red)!important;
    box-shadow:0px 0px 11px red !important;
}
.hpp-datepicker.datepicker-days td.active {
    background-color: #00A754 !important;
}
.hpp-nav > ul.nav > li.active > a::after {
    border-left-color:#00A754!important;
}
.hpp-loading-spinner {
    background: url('loading.gif') no-repeat; /* Caution: relative path is not allowed. To reach proper functionality, image source file has to be called from absolute path over HTTPS secure protocol.*/
}
.hpp-container .panel-heading {
    background: transparent none repeat scroll 0% 0%;
    -webkit-transition: background-color 0.4s;
    -moz-transition: background-color 0.4s;
    -ms-transition: background-color 0.4s;
    -o-transition: background-color 0.4s;
    transition: background-color 0.4s;
    cursor: pointer;
    cursor: hand;
}
.hpp-container .panel-heading:hover {
    background-color: #DBDBDB;
    -webkit-transition: background-color 0.4s;
    -moz-transition: background-color 0.4s;
    -ms-transition: background-color 0.4s;
    -o-transition: background-color 0.4s;
    transition: background-color 0.4s;
    cursor: pointer;
    cursor: hand;
}
/* ***** MAIN ELEMENTS THEMING - END ***** */
/* ***** ENABLING OPTIONAL FIELDS - START ***** */
.hpp-creditcard-form-row-for-street1-field {
    display : block;
}
```

```

.hpp-creditcard-form-row-for-street2-field {
    display : block;
}
.hpp-creditcard-form-row-for-city-field {
    display : block;
}
.hpp-creditcard-form-row-for-state-and-postalcode-fields {
    display : block;
}
.hpp-creditcard-form-row-for-country-field {
    display : block;
}
.hpp-creditcard-form-row-for-email-field {
    display : block;
}
.hpp-creditcard-form-row-for-phone-field {
    display : block;
}
/* ***** ENABLING OPTIONAL FIELDS - END *****/
/* ***** CUSTOMIZING CREDIT CARD PAYMENT PAGE - START *****/
/* Setting Optional text to italic and changing color to silver */
form#hpp-creditcard-form small[data-i18n~="optional"] {
    color: #D9D9D9;
    font-style: italic;
    font-weight: 900;
}
/* ***** ENABLING OPTION FIELDS - END *****/
/* ***** CUSTOMIZING SEPA PAYMENT PAGE - START *****/
/* Highlighting SEPA payment consent checkbox */
#sepaDirectDebitForm.form-horizontal .radio, #sepaDirectDebitForm.form-horizontal .checkbox {
    min-height: 27px;
    background: #EDEDED;
    padding: 8px;
}
/* ***** ENABLING OPTION FIELDS - END *****/

```

4 Payment mobile SDK

4.1 Preface

The basic principle is that Payment SDK offers a seamless user experience by rendering a payment form as an integral part of merchant's mobile application.

4.1.1 Supported payment methods and transaction types

Payment methods	Transaction types
credit card	authorization-only
	authorization
	capture-authorization
	purchase

	referenced-authorization referenced-purchase
Paypal	authorization-only
	authorization
	capture-authorization
	debit
	recurring transactions
SEPA Direct Debit	authorization
	pending-debit
	referenced-authorization
	referenced pending-debit
Apple Pay*	purchase

*iOS Environment exclusive

4.1.2 System Requirements

Requirements for Android

- Android 4.1 (Jelly Bean / API Level 16)
- device cannot be rooted
- ElasticEngine Card Payment configuration details – merchant account ID, Secret Key and endpoint URL

Requirements for iOS

- Computer running OSX
- XCode > 7.1.1
- Device running iOS > 7.1
- Wirecard's Card Payment configuration details – merchant account ID, Secret Key and endpoint URL

4.2 Payment SDK integraton

The Payment SDK integrator should use GitHub dependency manager for easy integration of the Payment SDK into their application.

The specific location with installation guide will be provided. Prior to using the SDK you must get in touch with Wirecard support to create the merchant account for you.

Wirecard support will provide you with:

- Merchant Account ID
- Secret Key
- Endpoint URL

4.2.1 Calculation of Signature

As payment information is exchanged between the merchant's system, the consumer's browser, and Wirecard's Payment Processing Gateway, it is important that the data exchange safeguards against man-in-the-middle attacks. Mobile payment endpoint uses a digital SHA-256 signature for all message exchanges. This digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature let a recipient believe that the message was created by a known sender, and that it was not altered, while transmitting.

4.2.1.1 Request Signature

Merchant needs to create a digital signature on his own server as it is the only place where the Secret is stored. The client's server also provides the Merchant Account ID to the mobile app client.

To calculate the signature, following conditions need to be met:

- Fields need to be concatenated,
- leading and trailing space removed,
- SHA-256 signature follows the combined string

Please note that 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).

1. `request_time_stamp`
2. `request_id`
1. `merchant_account_id`
2. `transaction_type`
3. `requested_amount` (must be in dot-decimal notation)
4. `requested_amount_currency`
5. [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.

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'
secret_key               = 'efabf47b-e43b-4785-873f-1c5bc65b7cd2'

Pre SHA-256 string
20120430123012order-12345b19fb056-d8da-449b-ac85-
cfbfd0558914purchase1.01USDefabf47b-e43b-4785-873f-1c5bc65b7cd2

SHA-256 signature
e93ed221efb8f6048df31794609d9557f65f175659e4928d10463b8998e3f61f

```

4.2.1.2 Secret Key Exchange

To ensure the authenticity of the request and response messages, it is required that a Secret Key is shared with the merchant. The Secret Key is used in generation of the `request_signature` fields (described in next chapter).

It is important that the Secret Key is never shared with anyone and is protected within the merchant server (only used in server side code for generating the request signature or validating the response signature).

Never store your secret key inside your application nor on a mobile device.

The Secret Key will be communicated at the time of Merchant Account setup. Please contact your support representative if you did not receive a Secret Key or require Secret Key regeneration.

4.2.1.3 Samples for signature code generation

Below you find code samples in various programming languages that you can use in your shop system. These samples take care of generating digital request signature.

PHP

```

$request_signature = hash('sha256', trim($request_time_stamp .
$request_id . $merchant_account_id . $transaction_type . $requested_amount
. $requested_amount_currency . $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 = 'XXXXXXXXXXXXXXXXXXXX';

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);

```

4.3 Android Environment

4.3.1 Credit Card Payment

The Payment SDK supports all major credit card schemes. Credit card transactions supported directly from the SDK:

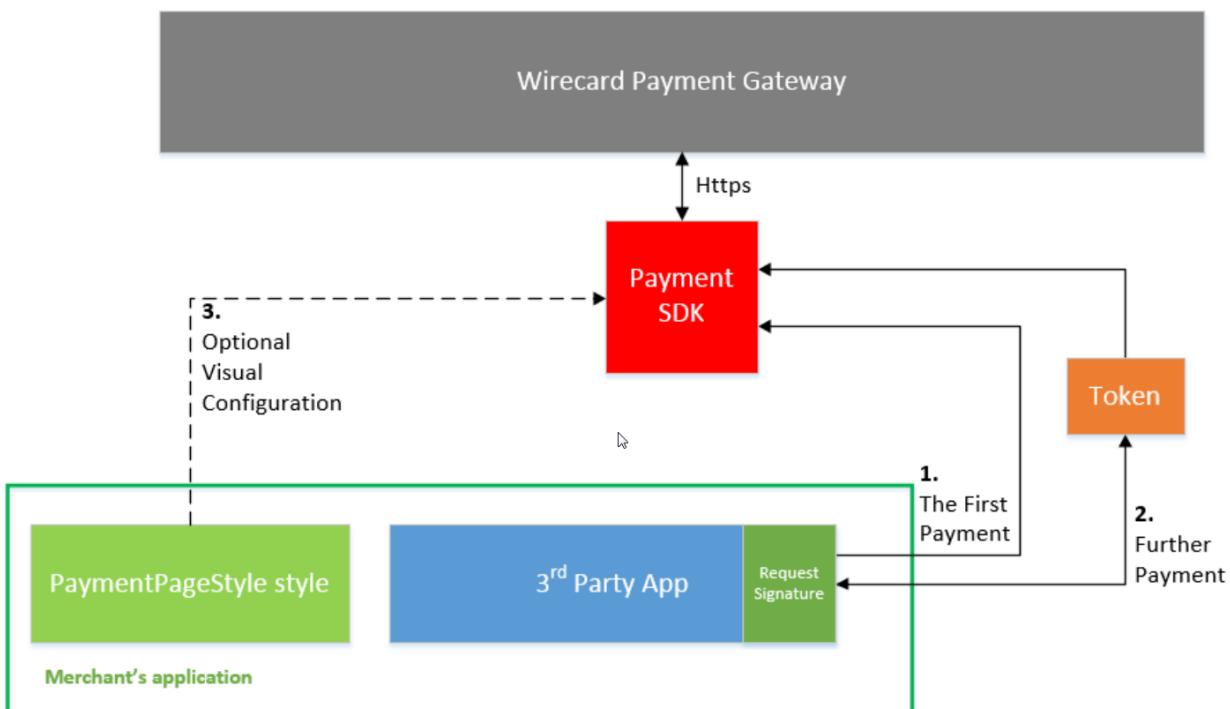
- **Purchase** - Takes funds from the [Card Holder]'s account. A one-step process to conduct two Transaction Types: [Authorization] and [Capture]. It can refer to a previous transaction only referred through a token-id.
- **Authorization-Only** - Verifies the card's validity without leaving an authorized amount.
- **Authorization** - Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture] on this transaction. It can refer to a previous transaction referred only through a token-id.
- **Capture-authorization** - Takes funds from the [Card Holder]'s account. Must follow an [Authorization] or [Authorization Supplementary] chain referred by a [Parent Transaction ID].
- **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 referred through a [Parent Transaction ID]. Amount can be amended.
- **Referenced-purchase** - Takes funds from the [Card Holder]'s account. Identical to a [Purchase] except for the fact that it refers to any previous purchase referred through a [Parent Transaction ID] with any amount.

Follow-up transactions like refund, credit and void are supported **only via REST API endpoint**:

- **Refund** - Gives funds to the [Card Holder]'s account, referring to an eligible previous [Capture], [Purchase] etc.
- **Credit** - Moves funds from the [Merchant] to the [Account Holder].
- **Void** - Frees reserved funds from the [Card Holder]'s account.

Usage of CVC is generally mandatory except for capture-authorization and is optional for referenced payments.

4.3.1.1 Functional flow



1. Merchant sends a calculated signature based on the Secret Key along with any payment request. Gateway then compares the signature and compares it with the one on its side. If the gateway accepts and processes the request, the response is sent back with the Token information.

2. Depending on your settings, any further request can be made using the Token without the need to enter the credit card information again (the Token would have to be stored on the 3rd party side – securely).
3. (optional step) 3rd party app initiates the Payment SDK with the optional `PaymentPageStyle` style object (used for styling the payment form provided by the SDK, see chapter [2.6 Styling of a Payment Form](#)).

4.3.1.2 Performing online card payments

The payment form contains four input fields:

- cardholder's name
- card number
- card validity and card security code
-

Customer confirms by hitting a PAY NOW button. When ABORT button is clicked, client is returned to an initial screen.

To perform a card payment, call the `makeCardPayment` method with following classes which needs to be initialized for the card payment process.

```
makeCardPayment(WirecardPayment payment, PaymentPageStyle style)
```

Attributes of the `WirecardPayment` class:

- `requestTimeStamp` [String]
- `requestID` [String]
- `merchantID` [String]
- `transactionType` [enum]
- `amount` [Decimal]
- `currency` [String]
- `requestSignature` [String]
- `tokenID` [String, optional]
- `parentTransactionID` [String, optional]
- `requireCVC` [Boolean, false by default for referenced payments]

Optional fields:

- `order`

- appendIPAddress
 - accountholder (billing address)
 - shippingAddress
 - Notifications
-)

4.3.1.3 Description of the API field

requestTimeStamp	timestamp of the request
requestID	Unique request ID generated by the merchant system
merchantID	Merchant ID of the merchant in Elastic Engine gateway
transactionType	Transaction type you wish to perform (initial transaction types: authorization-only, authorization, purchase)
amount	amount to charge the cardholder; note that the amount uses a dot for dividing integers and decimals (e.g. 1.03)
currency	currency code (e.g. EUR, USD)
requestSignature	signature calculated for this request by the merchant system
tokenID	card token which is created at each card payment and can be used by referencing the Token-ID instead of the Parent Transaction ID
parentTransactionID	a transaction Id to be used in capture and referenced transaction types
requireCVC	in the case of referenced transactions CVC entry may not be required by default – set flag to YES to override this behaviour. However, default implementation for non-referenced payments always requires input of CVC. For transaction type capture/authorization this attribute is ignored.

Optional fields:

order	order details of type Order
appendIPAddress	enabled merchant do include IP address of a device, default value ?true?
accountHolder	billing data of type CustomerData
shippingAddress	shipping data of type CustomerData
Notifications	array of Notification

```
Order class:
• Number
• Detail
• Description
• Descriptor
• Items - array of OrderItem

OrderItem:
• Name
```

```

• Description
• ArticleNumber
• Amount
• TaxAmount
• Quantity

Address: 13
• street1
• street2
• city
• state
• country
• postal-code

CustomerData:
• Email
• Phone
• firstName
• lastName
• Gender
• dateOfBirth
• Address type of address

Notification:
• transactionState
• url

```

4.3.1.4 Referenced payments

Transaction types: referenced-authorization, referenced-purchase

It is possible to place referenced transactions based on a previous transaction. Such transactions always refer to a parent transaction of the initial payment.

- **parentTransactionID** – Transaction-ID (also referred to as parent transaction) to be used in capture-authorization and referenced transaction types.
- **requireCVC** – for referenced payments entry of CVC can be deactivated for faster checkout



4.3.1.5 Token payments

Transaction types: authorization, purchase

tokenID – card token which is created at each card payment and can be used by referencing the Token-ID instead of the Parent Transaction ID. You can only place a transaction with token from any previous authorization or purchase transaction.



4.3.2 Sample for a makeCardPayment method call

The call consists of following three parts and in this case has no custom style applied:

```

paymentSDK = new PaymentSDK(context);
+
Select one option from a - c;
+
paymentSDK.makeCardPayment(payment, null);
  
```

```

paymentSDK = new PaymentSDK(context);

a) Simple initial payment (does not require tokenID, parentTransactionID, cvvRequired)

Payment payment = new Payment(timestamp, requestID, merchantID, transactionType, amount, currency, signature);

b) Referenced payment using Parent Transaction ID
Payment payment = new Payment(timestamp, requestID, merchantID, transactionType, amount, currency, signature, null, parentTransactionID, cvvRequired);

c) Payment using Token ID
Payment payment = new Payment(timestamp, requestID, merchantID, transactionType, amount, currency, signature, tokenID, null, cvvRequired);
paymentSDK.makeCardPayment(payment, null);
  
```

4.3.2.1 Styling of a payment form

You can adjust the look and feel of your payment form and buttons. However, you cannot change the position of any field or button. An example showcase how to create a

PaymentPageStyle object.

The call consists of following equal three parts with difference that a custom style is applied:

```

paymentSDK = new PaymentSDK(context);
+
Select one option from a - c;
+
paymentSDK.makeCardPayment(payment, style);
  
```

The actual style is defined like this:

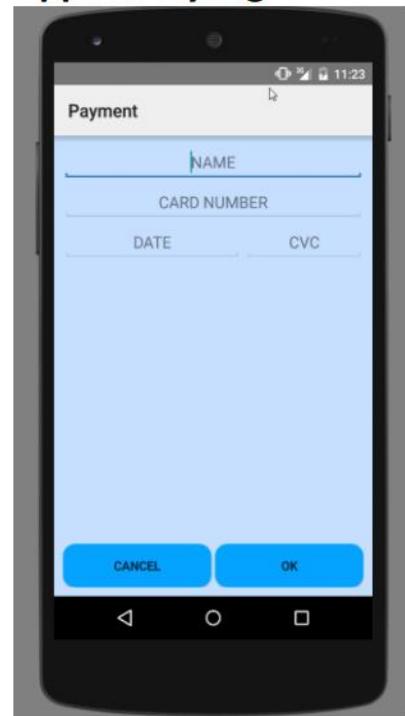
```
PaymentPageStyle style = new PaymentPageStyle();
style.cardNameHint = "NAME";
style.cardNumberHint = "CARD NUMBER";
style.cardCvvHint = "CVC";
style.cardDateHint = "DATE";
style.payButtonText = "OK";
style.cancelButtonText = "CANCEL";
style.loadingDialogMessageText = "PROCESSING TRANSACTION";
style.backgroundResourceId = R.color.background;
style.payButtonBackgroundResourceId = R.drawable.pay_button;
style.cancelButtonBackgroundResourceId = R.drawable.cancel_button;
style.inputBackgroundResourceId = R.drawable.edit_text_holo_light;
paymentsdk.makeCardPayment(payment, style);
```

An example of what can be styled in comparison to custom format is below:

Default styling



Applied styling



4.3.3 Paypal payment

4.3.3.1 Performing Paypal payment

Call the *makePayPalPayment* method to initiate the payment process.

```
makePayPalPayment(WireCardPayment wireCardPayment, WireCardPeriodicType
periodicType,WireCardSequenceType sequenceType)
```

periodicType	Indicates how and why a Payment occurs more than once. Possible values include: a) 'installment': one in a set that completes a financial transaction and b) 'recurring': one in a set that occurs repeatedly, such as a subscription.
sequenceType	Used in conjunction with periodicType to indicate the sequence. Possible values include: a) 'first': first transaction in a set. b) 'recurring': subsequent transactions in the set. c) 'final': the last transaction in the set.

Attributes of the wireCardPayment class:

```
WirecardSepaPayment:
• requestTimeStamp [String]
• requestID [String]
• merchantID [String]
• transactionType [enum]
• amount [Decimal]
• currency [String]
• requestSignature [String]
• tokenID [String, optional]
• parentTransactionID [String, optional]
• requireCVC [Boolean, false by default for referenced payments]

WirecardPeriodicType:
It can be one of: recurring, installment

WirecardSequenceType:
It can be one of: first, recurring, final
```

4.3.3.2 Sample for makePayPalPayment method call

```
WireCardPayment wireCardPayment = new WireCardPayment(timestamp, requestID,
merchantID,
transactionType, amount, currency, signature);
paymentsSDK.makePayPalPayment(wireCardPayment, null, null);
```

4.4 SEPA Direct Debit payment

4.4.1 Performing SEPA payment

Call the *makeSepaPayment* method to initiate the payment process.

```
makeSepaPayment(WirecardSepaPayment payment, PaymentPageStyle style,
WirecardPeriodicType
periodicType, WirecardSequenceType sequenceType)
```

payment	Payment data to submit to the Wirecard Payment Gateway
periodicType	Indicates how and why a Payment occurs more than once. Possible values include: a) 'installment': one in a set that completes a financial transaction b) 'recurring': one in a set that occurs repeatedly, such as a subscription
sequenceType	Used in conjunction with periodic_type to indicate the sequence. Possible values include: a) 'first': first transaction in a set b) 'recurring': subsequent transactions in the set c) 'final': the last transaction in the set.

Attributes of the `WirecardSepaPayment` class (SEPA specific objects are highlighted):

```
WirecardSepaPayment:
• requestTimeStamp [String]
• requestID [String]
• merchantID [String]
• transactionType [enum]
• amount [Decimal]
• currency [String]
• requestSignature [String]
• tokenID [String, optional]
• parentTransactionID [String, optional]
• requireCVC [Boolean, false by default for referenced payments]
• creditorId [String]
• mandateId [String]
• signedDate [Date]
• merchantName [String]
• dueDate [Date]

WirecardPeriodicType:
It can be one of: recurring, installment

WirecardSequenceType:
It can be one of: first, recurring, final
```

4.5 Response Handling

In order to return the call result you need to ensure to add `onActivityResult` method:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    PaymentResponseWrapper responseWrapper =
    paymentSDK.onActivityResult(requestCode, resultCode,
    data);
    if (responseWrapper != null) {
        if (responseWrapper.getPayment() != null) {
            // handleResult(responseWrapper.getPayment());
        } else {
            // handleErrorResult(responseWrapper.getError());
        }
    }
}
```

In this method you obtain a `PaymentResponseWrapper` object which contains:

```
PaymentResponse payment
ResponseError error

PaymentResponse class contains:

Statuses statuses
String transactionId
String transactionType
String requestId
MerchantAccountId merchantAccountId
TransactionState transactionState
```

```

long completionTimestamp
RequestedAmount requestedAmount
CardToken cardToken
AccountHolder accountHolder
String authorizationCode
PaymentMethods paymentMethods
String cancelRedirectUrl 20
String failRedirectUrl
String successRedirectUrl
CreditCard creditCard
ThreeD threeD
Notifications notifications
CustomFields customFields

```

ResponseError class contains:

```

int errorCode
String errorMessage

```

4.6 iOS Environment

4.6.1 Apple Pay

Apple Pay is a mobile payment and digital wallet service by Apple Inc. that lets users make payments using the iPhone 6, iPhone 6 Plus, Apple Watch, iPad Air 2, and iPad Mini 3. This method is configured in a processing model which means funds are collected by provider (Payeezy) which has a signed agreement with Apple to process payments with them.

4.6.1.1 Perform Apple Pay payment

For scenario “a)” use a class called **WDApplePayPayment** given a merchant’s developer himself implements **PKPaymentRequest**.

```

/**
@brief Defines ApplePay payment method.
@discussion It is merchant's app responsibility to gather parameters via
PKPaymentAuthorizationViewController
*/
@interface WDApplePayPayment : WDPayment
- (nullable instancetype)init NS_UNAVAILABLE;
/** 
@brief Initializes payment object with parameters gathered via
PKPaymentAuthorizationViewController
@param payment PKPayment object gathered via
PKPaymentAuthorizationViewController
@param summaryItems summaryItems gathered via
PKPaymentAuthorizationViewController
@param currency currency code. Supported codes: @"EUR", @"USD"
@return initialized object with payment data
*/
- (nullable instancetype)initWithPayment:(nonnull PKPayment *)payment
summaryItems:(nonnull NSArray<PKPaymentSummaryItem
*> *)summaryItems
currencyCode:(nonnull NSString *)currency
NS_DESIGNATED_INITIALIZER;
@end

```

Another method is when Payment SDK automatically implements apple PKPaymentRequest and its methods. The PaymentSDK offers the **WDApplePayPayment** class where the developer needs to specify the merchant ID set-up in the step 2 of the pre-requisites section.

```
/**
@brief Defines ApplePay payment method.
@discussion WDClient handles PKPaymentAuthorizationViewController. As it does
not support complex use cases (shipping address selection, etc.) it is
intended for simple payments. In case of complex use cases handle
PKPaymentAuthorizationViewController by your own and use WDApplePayPayment
*/
@interface WDApplePayManagedPayment : WDApplePayPayment
/**
@brief Apple Merchant ID
*/
@property (strong, nonatomic, nonnull) NSString *appleMerchantID;
/**
@brief Merchant Country
*/
@property (assign, nonatomic) WDCountry merchantCountry;
/**
@brief prompt user for billing address
*/
@property (assign, nonatomic) BOOL requiredBillingAddress;
/**
@brief prompt user for shipping address
*/
@property (assign, nonatomic) BOOL requiredShippingAddress;
/**
@brief initialize object
@param appleMerchantID Apple Merchant ID
@param merchantCountry merchant's country
@return initialized object
*/
- (nullable instancetype)initWithMerchant:(nonnull NSString *)appleMerchantID
andCountry:(WDCountry)merchantCountry
NS_DESIGNATED_INITIALIZER;
- (nullable instancetype)initWithPayment:(nonnull PKPayment *)payment
summaryItems:(nonnull NSArray<PKPaymentSummaryItem
*> *)summaryItems
currencyCode:(nonnull NSString *)currencyCode
NS_UNAVAILABLE;
@end
```

4.6.1.2 Pre-requisites

1. Merchant needs to register with Payeezy and create a merchant. Registration can be either using Lite registration for testing or Certified Live registration – please follow the Apple Pay developer guide <https://goo.gl/TJdPmp>
2. Register the merchant ID obtained from Payeezy in your Apple Developer account – please follow the Apple Pay developer guide <https://goo.gl/67GBtv> and setup your XCode project according to the instructions.
3. Merchant needs to register for Wirecard Payment Processing API in order to get an active merchant account. Payeezi credentials (API Key, API Secret, Merchant Token, Merchant App Label) must be forwarded as the local Wirecard's account must be paired.
 - API credentials: API Key, API Secret
 - Merchant credentials: Token, App Label

4. A valid Credit or Debit card issued by a bank participating in Apple Pay scheme must be at hand – please see a list of participating banks.

Supported transaction type is purchase only and there are 2 flows based on users requirements:

- a) If a merchant requires a full control over PKPaymentRequest* (dynamic shipping methods and pricing) then use WDApplePayPayment.
- b) If the purchase is simple with fix shipping pricing use WDApplePayManagedPayment to handle PKPaymentRequest by SDK.

* The PKPaymentRequest class encapsulates a request for payment, including information about payment processing capabilities, the payment amount, and shipping information.

4.6.2 Credit Card

4.6.2.1 Functional Flow

Please see: 4.3.1.1

4.6.2.2 Perform online card payments

The payment form contains four input fields:

- cardholder's name
- card number
- card validity
- card security code

Customer confirms by hitting a PAY NOW button. When ABORT button is clicked, client is returned to an initial screen.

To perform a card payment, call **[WDClient makePayment:withCompletion:]** method with payment object of type **WDCardPayment**.

```
/**
 * @brief Trigger payment process
 *
 * @param payment defines payment process
 * @param completionBlock will be called at the very end of payment flow. It
 * provides payment response or a descriptive error with more details in the
 * payment response
 */
- (void)makePayment:(nonnull WDPayment *)payment
withCompletion:(nonnull WDCompletionBlock)completionBlock;
```

WDCardPayment :

```
/** Defines Card payment method.
*/
@interface WDCardPayment : WDPayment
/** mark payment as recurring */
@property (assign, nonatomic, getter=isRecurring) BOOL recurring;
/** transaction identifier of previous transaction gathered by
[WDPaymentResponse transactionIdentifier]
@discussion Usually used with transaction types:
WDTxTypeCaptureAuthorization,
```

```

WDTransactionTypeReferencedAuthorization and
WDTransactionTypeReferencedPurchase
*/
@property (strong, nonatomic, nullable) NSString *parentTransactionID;
15
/**
@brief tokenized card data gathered by previous transaction
[WDPaymentResponse cardToken].
@discussion set [WDCardToken tokenID] if you want to use the same card as in
previous transaction. In case of transaction referenced by parentTransactionID
and security code is required set [WDCardToken maskedAccountNumber] to help
user with security code typing
*/
@property (strong, nonatomic, nullable) WDCardToken *token;
/**
@brief Initialize Car payment object
@param amount Amount that accompanies the transaction
@param amountCurrency Currency in which the transaction is processed
@param transactionType WDTransactionType transaction type
@return a newly initialized object
*/
- (nullable instancetype)initWithAmount:(nonnull NSDecimalNumber *)amount
amountCurrency:(WDCurrency)amountCurrency
transactionType:(WDTransactionType)transactionType;
@end

```

4.6.2.3 Referenced payments

Please see: 4.3.1.4

4.6.2.4 Token payments

Please see: 4.3.1.5

4.6.3 Description of the API fields

All transactions are handled by class WDClient:

```

/**
Payment processing client
*/
@interface WDClient : NSObject
- (nullable instancetype)init NS_UNAVAILABLE;
/**
* @brief It initializes client for processing payments.
*
* @param environment Environment for processing payments
*
* @return An initialized object, or nil if an object could not be created for
some reason that would not result in an exception
*/
- (nullable instancetype)initWithEnvironment:(WDEnvironment)environment
NS_DESIGNATED_INITIALIZER;
/**
* @brief Trigger payment process
*
* @discussion Payment method is defined by payment object. Each payment
method is defined in separate subclass of WDPayment. Supported payment methods:
WDApplePayManagedPayment, WDApplePayPayment, WDCardPayment, WDPayPalPayment,
WDSEPAPayment

```

```

/*
 * @param payment defines payment process
 * @param completionBlock will be called at the very end of payment flow. It
provides payment response or a descriptive error with more details in the
payment response
*/
- (void)makePayment:(nonnull WDPayment *)payment
withCompletion:(nonnull WDCompletionBlock)completionBlock;
/**
 * @brief Trigger payment process
 *
 * @discussion Payment method is defined by payment object. Each payment
method is defined in separate subclass of WDPayment. Supported payment methods:
WDApplePayManagedPayment, WDApplePayPayment, WDCardPayment, WDPayPalPayment,
WDSEPAPayment
*
 * @param payment defines payment process
 * @param style not used at the moment
 * @param completionBlock will be called at the very end of payment flow. It
provides payment response or a descriptive error with more details in the
payment response
*/
- (void)makePayment:(nonnull WDPayment *)payment
withStyle:(nullable WDStyle *)style
withCompletion:(nonnull WDCompletionBlock)completionBlock;
@end

```

Base model object of each payment method is WDPayment class:

```

/** Base class of all Payment Methods.
Initialization WDPayment object is unavailable. Please use derived classes for
initialization.
*/
@interface WDPayment : NSObject
/**
@brief Authorize client to process the transaction.
@discussion TODO: describe process how to generate signature. It is mandatory.
*/
@property (strong, nonatomic, nonnull) NSString *requestSignature;
/**
@brief Date when requestSignature was generated
@discussion It is mandatory.
*/
@property (strong, nonatomic, nonnull) NSDate *requestTimestamp;
/**
@brief Unique identifier assigned for every Merchant Account.
@discussion It is mandatory.
*/
@property (strong, nonatomic, nonnull) NSString *merchantAccountID;
/**
@brief Unique identifier associated with the transaction, which is created by
the merchant.
@discussion It is mandatory.
*/
@property (strong, nonatomic, nonnull) NSString *requestID;
/**
@brief The only amount that accompanies the transaction when it is created
and/or requested.
@discussion For transactionType WDTransactionTypeAuthorizationOnly the only
valid value is [NSDecimalNumber zero].
It is mandatory.
*/
@property (strong, nonatomic, nonnull) NSDecimalNumber *amount;
/**
@brief Currency in which the transaction is processed.

```

```

@discussion It is mandatory.
*/
@property (assign, nonatomic) WDCurrency amountCurrency;
/**
@brief Determines transaction processing behaviour.
@discussion It is mandatory.
*/
@property (assign, nonatomic) WDTransactionType transactionType;
/**
@brief Customer's account information.
@discussion It is optional.
*/
@property (strong, nonatomic, nullable) WDCustomerData *accountHolder;
<1>
/**
@brief Customer's shipping information
@discussion It is optional.
*/
@property (strong, nonatomic, nullable) WDCustomerData *shipping;
/**
@brief Customer's order information.
@discussion It is optional.
*/
@property (strong, nonatomic, nullable) WDOrder *order;
/**
@brief Notifications configuration
@discussion It is optional.
*/
@property (strong, nonatomic, nullable) NSArray<WDNotification *>
*notifications;
/**
@brief The IP Address of the Customer as recorded by the entity receiving the
Transaction Attempt from the Customer
@discussion It is optional.
*/
@property (strong, nonatomic, nullable) NSString *IPAddress;
/**@brief Defines the user's language and any special variant preferences that the user
wants to see in their user interface. @discussion Typically used by WDPayPalPayment and 3D
Secure WDCardPayment HTMLformulars. It is optional.
*/
@property (assign, nonatomic) WDLocale locale;

- (nullable instancetype)init NS_UNAVAILABLE;

/**
@brief Returns the notification for transaction state.
@discussion@param state The transaction state defined by WDTransactionState enumaration
@return Returns notification or nil if no associated notification.*/
- (nullable WDNotification *)notificationForState:(WDTransactionState)state;

@end

```

4.7 Paypal for iOS

PayPal Express Checkout is a service that allows customers to pay using a PayPal account instead of a credit card. Customers can make purchases without entering billing, shipping, and credit card information as the information is already stored with PayPal. Merchants should provide details about the order so that the order information can be displayed to the customer during the PayPal checkout process.

4.7.1 Perform Paypal payment

Call **[WDClient makePayment:withCompletion:]** method with **WDPayPalPayment** object:

```
/** Defines PayPal payment method.
```

```
/*
@interface WDPayPalPayment : WDPayment
/**
@brief Transaction ID of the first transaction in the series.
@discussion Mandatory for "recurring" and "final" transactions
*/
@property (strong, nonatomic, nullable) NSString *parentTransactionID;
/**
@brief It describes recurring transactions.
*/
@property (strong, nonatomic, nullable) WDPeriodic *periodic;
/**
@brief Initialize WDPayPalPayment object
@param amount amount that accompanies the transaction
@param currency currency in which the transaction is processed
@return a newly initialized object
*/
- (nullable instancetype)initWithAmount:(nonnull NSDecimalNumber *)amount
currency:(WDCurrency)currency NS_DESIGNATED_INITIALIZER;
21
@end
```

4.8 SEPA

SEPA is an abbreviation for Single European Payment Area. The main goal of SEPA is to make the process of transferring money from one bank account to another more easy. With SEPA a Transfers of Funds (e.g. Deposits, Credits and Debits) for all european countries that do business in EURO currency should be easy and over the country boarders unique and standardised way.

4.8.1 Perform a SEPA payment

Call **[WDClient makePayment:withCompletion:]** method with **WDSEPPayment** object:

```
/** Defines SEPA payment method.
*/
@interface WDSEPPayment : WDPayment
/**
@brief The Creditor Id for the Merchant Account SEPA.
@discussion It is mandatory.
*/
@property (strong, nonatomic, nonnull ) NSString *creditorID;
/**
@brief The Mandate Id for the Mandate Transaction, generated by merchant.
*/
@property (strong, nonatomic, nonnull ) NSString *mandateID;
/**
@brief The date that the Mandate was signed.
@discussion If value is nil, current date will be used. It is optional.
*/
@property (strong, nonatomic, nullable) NSDate *mandateSignedDate;
/**
@brief The date that the Mandate Transaction is due
*/
@property (strong, nonatomic, nullable) NSDate *dueDate;
23
/**
@brief transaction identifier of previous recurring transaction gathered by
[WDPaymentResponse transactionIdentifier]
*/
```

```

@property (strong, nonatomic, nullable) NSString *parentTransactionID;
/**
@brief It describes recurring transactions.
*/
@property (strong, nonatomic, nullable) WDPeriodic *periodic;
- (nullable instancetype)initWithCreditor:(nonnull NSString *)creditorID
andMandate:(nonnull NSString *)mandateID
NS_DESIGNATED_INITIALIZER;
@end

```

4.9 Test Credentials

Test Credit Cards credentials

type	credit card number	Valid	CVV
non-3D card	4444333322221111	any	any
3D card	4012000300001003	01/2019	003

SEPA credentials

IBAN	DE83203205004989123456
BIC	DABADEHHXXX

Paypal credentials

Once redirected to Wirecard authentication screen:

username	buyer@wirecard.com
password	Einstein35

5 Accessing Resources

5.1 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.

5.2 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.

5.3 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.

6 Submitting Transactions

6.1 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.

6.2 Create a Payment

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

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

6.2.1 Fields

The following elements are mandatory (X) / optional (O) for sending a request/response for the payment method **creditcard**.

Field	Request	Response	Notification	Datatype	Size	Description
payment	X	X	X			This is a transaction.
merchant-account-id	X/O	X	X	String	36	Unique identifier for a merchant-account.
merchant-account-resolver-category	X/O	X	X	String	36	The category used to resolve the merchant-account based on rules on input.

transaction-id		X	X	String	36	This is the unique identifier for a transaction. It is generated by EE.
request-id	X	X	X	String	150	This is the identification number of the request. It has to be unique for each request.
transaction-type	X	X	X	String	30	This is the type for a transaction.
transaction-state		X	X	String	12	This is the status of a transaction
completion-time-stamp		X	X	Timestamp		This is the time-stamp of completion of request.
avs-code		O	O	String	36	This is the result of address's validation
csc-code		O	O	String	36	This is the result of CVC validation
statuses		X	X			This is the status of a transaction.
statuses.status		X	X			This is the status of a transaction.
status@code		X	X	String	12	This is the code of the status of a transaction.
status@description		X	X	String	256	This is the description to the status code of a transaction.
status@severity		X	X	String	20	This field gives information if a status is a warning, an error or an information.
requested-amount		X	X	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
requested-amount@currency	X	X	X	String	3	This is the currency of the transaction.
parent-transaction-id	X/O	X/O	X/O	String	36	This is the unique identifier of the referenced transaction. This might be mandatory if "merchant-account-id" or "merchant-account-resolver-

						category" is not used
account-holder	X	X	X			
account-holder.first-name	X	X	X	String	32	This is the first name of the end-consumer.
account-holder.last-name	X	X	X	String	32	This is the last name of the end-consumer.
account-holder.email	O	O	O	String	64	This is the end-consumer's email-address.
account-holder.gender	O	O	O	String	1	This is the end-consumer's gender.
account-holder.date-of-birth	O	O	O	Date	0	This is the end-consumer's birth date.
account-holder.phone	O	O	O	String	32	This is the phone number of the end-consumer.
account-holder.address	O	O	O			This is the end-consumer's address.
account-holder.address.block-no	O	O	O	String	12	This is the block-no of the end-consumer
account-holder.address.level	O	O	O	String	3	This is the level of the end-consumer
account-holder.address.unit	O	O	O	String	12	This is the unit of the end-consumer
account-holder.address.street1	X	X	X	String	128	This is the first part of the end-consumer's street.
account-holder.address.street2	O	O	O	String	128	This is the second part of the end-consumer's street.
account-holder.address.city	X	X	X	String	32	This is the end-consumer's city.
account-holder.address.state	O	O	O	String	32	This is the end-consumer's state.
account-holder.address.country	X	X	X	String	3	This is the end-consumer's country.
account-holder.address.postal-code	O	O	O	String	16	This is the end-consumer's postal code.
account-holder.merchant-crm-id	O	O	O	String	64	This is the merchant-crm-id of end-consumer.
account-holder.social-security-number	O	O	O	String	14	This is the social security number of the end-consumer.
shipping	O	O	O			This is used for shipping information

shipping.first-name	X	X	X	String	32	This is first-name from shipping information
shipping.last-name	X	X	X	String	32	This is first-name from shipping information
shipping.phone	O	O	O	String	3	This is first-name from shipping information
shipping.address	X	X	X			This is used for specify the address from shipping information
card	X					This is used for credit card of the end-consumer
card.account-number	X/O			String	36	This is the card account number of the end-consumer. If is mandatory if "card-token" is not used
card.expiration-month	X			numeric	2	This is the card's expiration month of the end-consumer
card.expiration-year	X			numeric	4	This is the card's expiration year of the end-consumer
card.card-security-code	X/O			String	4	This is the card's security code of the end-consumer. Depending on configuration it might be mandatory.
card.card-type	X			String	15	This is the card's type of the end-consumer
card.track-1	O			String	256	This is the card's track-1 of the end-consumer
card.track-2	O			String	256	This is the card's track-2 of the end-consumer
card.card-emv	O					This is used for EMV data for credit card of the end-consumer
card.card-pin	O					This is used for PIN data for credit card of the end-consumer
card-token	X/O					This is used for card-token of the end-consumer

card-token.token-id	X/O	X	X	String	36	This is the token corresponding to "card.account-number" of the end-consumer. It is mandatory if "card.account-number" is not specified. It is unique on instance of EE.
card-token.masked-account-number	O	X	X			This is the masked version of "card.account-number" of the end-consumer. E.g. 440804*****7893
card-token.cryptogram	O	O	O			This is the cryptogram of corresponding to "card.account-number" of the end-consumer
ip-address	O	O	O	String	15	The global (internet) IP address of the consumers computer.
order-number	X	X	X	String	64	This is the order number of the merchant.
order-detail	O	O	O	String	65535	This is a field for details of an order filled by the merchant.
order-items	O	O	O			This is a field for order's items filled by the merchant.
descriptor	X	X	X	String	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', ',', '+', ',', '-' , ''
notifications	O	O	O			This is used for IPN (Instant Payment Notification)
notifications.notification	O	O	O			This is used for IPN (Instant Payment Notification)

notifications.notification@transaction-state	O	O	O	String	12	This is the status of a transaction when IPN will be sent.
notifications.notification@url	O	O	O	String	256	The URL to be used for the IPN. It overwrites the notification URL that is set up in the merchant configuration.
custom-fields	O	O	O			This is used for adding custom information related to transaction
custom-fields.custom-field	O	O	O			This is used for adding custom information related to transaction
custom-field@field-name	O	O	O	String	36	This is the name for the custom field.
custom-field@field-value	O	O	O	String	256	This is the content of the custom field. In this field the merchant can send additional information.
payment-methods	X	X	X			This is used for specifying the payment method used for this transaction
payment-methods.payment-method	X	X	X			This is used for specifying the payment method used for this transaction
payment-methods.card-types.card-type	O	O	O	String	15	This specify the types of card supported for this payment-method
payment-methods.payment-method@name	X	X	X	String	15	This is the name of the payment method that that is chosen from the end-consumer. Value "creditcard" should be used.
payment-methods.payment-method@url	O	O	O	String	256	The URL to be used for proceeding with payment on provider side.

authorization-code	X/O	X/O	X/O	String	36	the authorization-code can be 1. input for "capture" without reference on "authorization" 2. output for "authorization"
three-d	X/O	X/O	X/O			information about three-d-secure
pares	X/O	X/O	X/O	String	2048	encrypted result of three-d-secure check
eci	X/O	X/O	X/O	String	2	result of three-d-secure check
xid	X/O	X/O	X/O	String	36	id of three-d-secure check
cardholder-authentication-value	X/O	X/O	X/O	String	1024	proof of three-d-secure check
cardholder-authentication-status	X/O	X/O	X/O	String	1	status of three-d-secure check
pareq	X/O	X/O	X/O	String	2048	encrypted request of three-d-secure check
acs-url	X/O	X/O	X/O	String	1024	url of three-d-secure acs
cardholder-authentication-status	X/O	X/O	X/O	String	1	status of three-d-secure check
airline-industry	O	O	O			information about airline-industry
cruise-industry	O	O	O			information about cruise-industry
consumer-id			X	String	50	The id of the consumer.
api-id			X	String	36	The api-id is always returned in the notification.
processing-redirect-url	O	O	O	String	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.
cancel-redirect-url	X	X	X	String	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.

success-redirect-url	X	X	X	String	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	X	X	X	String	6	Code of the language. Can be any of CZ, DA, EN, DE, ES, FI, FR, IT, NL, PL, GR, RO, RU, SV and TR. Can be sent in the format <language> or in the format <language_country>.
entry-mode	O	O	O	String	24	This is information about the channel used for this transaction Can be one of the following: mail-order, telephone-order, ecommerce, mcommerce, pos
periodic	O	O	O	String	24	This is information about the periodicity of this transaction Can be one of the following: installment, recurring
signature			O			The Signature info, consisting of SignedInfo, SignatureValue and KeyInfo

6.2.2 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>
```

6.2.3 Instant Payment Notifications

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	15
bank-account.iban	M - for iDEAL	Alphanumeric	34
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	256
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

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>
```

6.2.4 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>

```

6.2.5 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&acc
ount_number=444433332221111&expiration_year=2013&card_security_code=123&payment_ip_a
ddress=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_c
ountry=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

```

6.2.5.1 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&account_number=4444333322211111&expiration_year=2020&card_security_code=123&payment_ip_address=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="201.000" 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>
```

6.2.5.2 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&account_number=4444333322221111&expiration_year=2020&card_security_code=123&payment_ip_address=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
    }
}
```

6.2.5.3 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&account_number=44443332221111&expiration_year=2020&card_security_code=123&payment_ip_address=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=201.0000&status_severity_1=information&token_id=4851746269541111&transaction_state=success&transaction_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=44443*****1111&status_description_1=3d-acquirer%3AThe+resource+was+successfully+created.&provider_transaction_id_1=C827886140412772968790&authorization_code=801397&
```

6.2.5.4 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=purchase&expiration_month=12&account_number=44443332221111&expiration_year=2013&card_security_code=123&payment_ip_address=127.0.0.1&last_name=Doe&email=test@test.com&phone=&address_street1=Einsteinring
```

```
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>

```

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

6.3.1 Fields

merchant-account-id	M	alphanumeric	36
transaction-id	M	alphanumeric	36

6.3.2 Sample Request

```
https://hostname/engine/rest/merchants/ba261be8-af94-11df-ab7800163e5eaf7/payments/048b27e0-9c31-4cab-9eab-3b72b1b4d498
```

6.3.3 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.

6.4 Retrieve Transaction by Request Id

This returns a single transaction belonging to a merchant account. The request id attribute must match 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	<code>https://hostname/engine/rest/merchants/{merchant-account-id}/payments/search?payment.request-id={request-id}</code>
Request Formats	XML, NVP
Response Formats	XML, HTML, JSON
Request Methods	GET

6.4.1 Fields

merchant-account-id	M	alphanumeric	36
request-id	M	alphanumeric	150

6.4.2 Sample Request

<https://hostname/engine/rest/merchants/ba261be8-af94-11df-ab78-00163e5eafed7/payments/search?payment.request-id=048b27e0-9c31-4cab-9eab-3b72b1b4d498>

6.4.3 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.

6.5 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.

6.5.1 Fields

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

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

6.5.2 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>

```

6.6 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.

6.6.1 Fields

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

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

6.6.2 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>

```

7 3-D Secure

7.1 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.

7.2 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.

7.2.1 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.

7.2.2 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.

7.3 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.

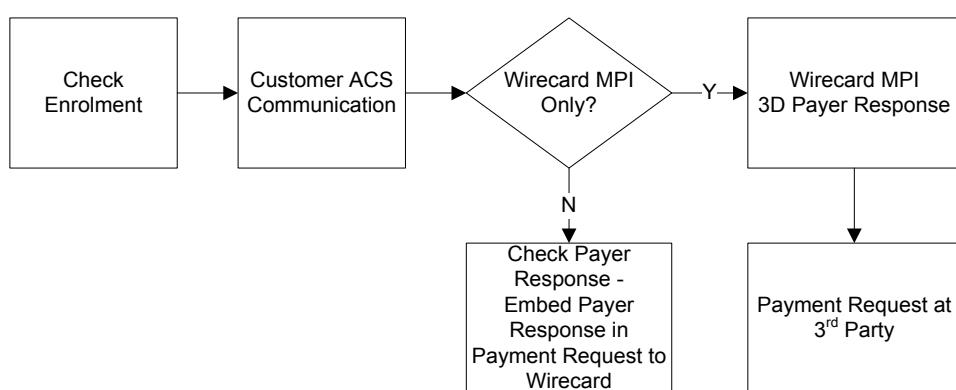


Figure 1: 3-D Secure Process Flow

See Section 7.4 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.

7.4 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.

7.4.1 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.

three-d.pareq	M	alphanumeric	16000
three-d.acs-url	M	alphanumeric	100

7.4.2 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>eJxtk9t0g0AQhu99CsK93RNQ2ixrqrUeEo22VeM1XSYtCScXsK1P77LFQwsQEmb2g/33nx1+sUsT6xNUGedZYJMBti3IZB7F2TqwX5azc9+2yirMojDJMwjsPZT2hTjjy40CmC5A1grEmaUv/gB1Ga7BiqPADhxMPDIc+T7DdEQZdnzPPnCGFZrM4eMvNr1WhdAiBpSjn/AYegA1N2FWHafNUig/Lu8eBaHMcT202rDLpaDupsLFGBP9YI40iS6YhSmIt1iBDfvkTSHNrcUmLzgyC11e5nVWqb3wqd7+J+hitUrEpqqKMULb7XawbTcYyDz1qFk8PjHqPzJ/qpt02SdkF0ditZw/z2dRsUiuyVvyej0/2b1Ldr9ffb0EHDVE97sorEBQTCh2qWsRd8zwGD00TL7H8LQ5Y1Mv7fbhvQsVrcxJCzeGn+R6nKyV0m2orXQ0/ht1QdgVui31P3TD/L6f+NdvFL+67e0hWeleGHR6pkPX9R3iMkoJHTHfdxjhnuksA/WqiXXJyQgTIyc+rT9H/3fV0v7GoKzmR89XOh4ur4B8cjnkw==</pareq>
      <acs-url>https://3dsecure-test.wirecard.com/acssim/app/bank</acs-url>
    </three-d>
  </payment>

```

7.5 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).

7.5.1 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

7.5.2 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>
```

7.5.3 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).

7.6 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.

7.6.1 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

7.6.2 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+jdvfp6eHem6quT1a+rCnf
ypL5P12aDJugK0M8e37CfqNPwyDzcj/OwucnXVv+mj79s5hrUREErBp4dREsBkM451JQ1m4YDGP/+YmczTCCo
GbYeDKZYegYw2bo0wN3x27BLijvSBQdkxiBzaZjbEa0J/h09g15R796s4D0/MbnyNvyM0gKCi9ys+qz+L7leh
dakBcYT0zJyRx5XX7HpUEhsAsSRVEM/qFz5CH4DszcNFiYcRF4buEP2SDNh2qUn+fIfem73svrrCquiykOzb8
tvsPqI11EVXX+gyBt2/5uXw389vJ0jtW2P0eM/BzyfFvfxoVPjnSxvzAIWXJ6+eriMmkkES6Zu6We5bh8BM9z
51b4fs53q2CBoxi0kjg5xCZ/UPIPis2Ru/yHhKe3GG/3BbP9mH8HnV/dbK/gW8K/yH7IZF0UkI4wlWMIf199B
wbd0c8CqAM5n3+JX8/J2p+drNF92FAzwjomvuCzr1g90qTn/M1F3+HV5Wb1WXC3u0vM5+CN1tmgUAgGaRzw
M8Bry00+SHHHjxAiVh/PD/z3pBEuzFXExpLU2fBV9y9TFYml1b/b7WvRqHGXS9CIbw5cjK56ePFCZ+50WI4LC
aEHSGQIBfxuFFt4v7qcAXsk0+mDNulmex5yZx71awsqWginJ/+07LTyq13U0rhuv45hdU+8vDxtmvwmQlMPIv
Ex5j8jSF114+IYt/ffxvVH/1ti jdX2XkYjdFu+AQ3CgXDPWd8Pz018/P11wr3Kw85EVafpj/b1aDrAmS/Bz4v
8o35280IB9Vs3EY1NX/E9NbPA8NhpvUwYJi4/021LsC10t8vKHaptppn+X6pHmQYx+Rc+Q9D3D+8Srfe/0AXo
1Gvcy62TYPFXu/Dlyfv0ZHqyt228rVkvqY4gBJsEuyr5gSWCdxE60Jed0Y/IwlkheZtxi9ZA/4NkrUwcg7bAo
72o9scflyTUuBTu2XU487+zNDb9GDZiX9aYaxbYG0skmORSphEHKyF9HR6Jig1ng1G4IxihGCnAfrfJKMSbRy
8pBmizVjkT1kFm30paTX1IpkrV1B7GYID7DMpPjrGYZUh1KrU1MvoCuVUQacGqCUX9uiMbBND7QNNm4s68Qip
1KWM1bahwhvw4NQbIAillqX7L24uzg+X+G6r3SWJVv4TmwI71qc15RxXroradTuB6wfLxUXB+M0Q+nLniqyID
1ABSSiLZXgINrrzQsFnp8fif+Q7Pk6uD5uwSLRGetW7m0m1vtj4FXyrTcx8jPh159ay9/Djf783swAD9fP5Ji
YoMQfFb5JwZB2s9MwPwwlt4b8iuty+FL5fw+ZZ0mfI18N3C0yQVHF1jRsDtIgsB5PcMAR5BK9AgFBhhbes
Yovr3BGixpOBwi1pBbSh03ZMD0Q61A0a2BpIDE1SuXbV2qyhKOsB10WybcmoY4msbYkngUtqnzeu+3SJuuast
s22k46gfSgotbWORR6x57yTHMpEqomeTtb29buPNjjZLRnaM0xZcxLDxaP7xKB85N9DDpJs1sZ3K2y3BX7bh
QpJzZhXB1dbhxeDxVT7ga+mVwdSw1woj3hHhyTTLyb5sodD/yZIm1owfcWNKExtK43rzJNKmVegHKQC9r0mo
e6UJSynbAKHfLPNeKsqFxhcScIBF1jomkwNaUHrbqfpPIKAwDeixG+7QM97gdauYSRpKgrmX0AtehDsvpA4kW
Hoc76cXAoZfmIxek7nW8BqxHrqBPrHzryiThWmJi48urw5DHPY62YcjFEkB5Rr3wA1XYE6zC0UDRARgLNNuCG
2ANcnizCqs1HTndHSfICx53R1vNukNv9kQjVTi509FEX1YncW+uMHd7cAaFz2+TaMLn+io6Bbv22GahjDJSma
3Vm0G1UTAGtKgJLyHRxAlFg6kk6ecpaYU4v3TaF1h1t6P1+aVnA7w6fVkcRvbJelispIWIVI7pL7ZEQSx+N
1Q1Mro8TsZR9tNW7ZN4wqiEY2vcS9jng609qE/uxUAQ62sQ2b6UNEr7Ivp+bDwyVwuiMhL7zbL1N+qmYY8H
7Vh2Z7ew9WQnLcena66fs9ySQpWq5Ehcp74mLJmcNqpBvNIZndnsE2PUsIqyrCXgrIUDzZ94PwScwqfbw9agJ
sfpcirW48Cj0AN1tGhmqrSpqqJ9aTQQsjQA/HEQBumWPstTtnsI7VKOB0AIWbG63ufKmNDhM4e1IDD0GLdfeMQ
kn2pYJbWhd2jSt6CtYhm24D/kBpIUvtCpdsgxduDx9nqj0Gc5zjXuRwINzdCQxhiF1ggYOD6rkGs99L6vBW13
dywoWjcSCN96dITH7W50qd17R4j6Dh9NbhsQRN3ya0kNdrhRCyzXbXpAvhFUOn22quPG0YdkVfDZyVFpdY/P
UITGEy/bnZ000T6cGYcDxTJQ159dXWtH3sj6xuY3Mt0vZAYKNEYIJpfnzG7VrukZUWX20Svp1Wb203zrbPUBm
MB0o0+IcwSf15zaHzgg2xY1omm9xDPU2ZKmhXm4qsr1GPWF30LkcqOPSpoc8ScPDdWuPqn+KexVsK4G6joW90
6M1J6xchqE3p7XXU7bK+/iUFsaFFAXT+oAwoaVvNqbRR1h8v1onh06zNIIsRIvByTLei/IyeI0kOXY0qly3Fx
wQGIXbNZ0Jnrow21Mwc7nTB3noE0tKX3p1xzPtEad7PwtZ1HAwXCEzf1DL1MpIzTwoVYHRir1kwa1ZgYTkVaw
PdTAVbd5N1udcXojo3qnURuZ2oV1YDa1MdqWE43rpqsinR1NxKZaNRcidTC+7AbZGQiPFrFx87wkDz6FPLeu
/7taovbF879m3IxgL97Pn5x/geuCnn</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>V3NMZzNya2N5Vlh2MWRFUmo2NjA=</xid>
            <cardholder-authentication-value>AAABD//////////AAAAAAA=</cardholder-authentication-value>
            <cardholder-authentication-status>Y</cardholder-authentication-status>
        </three-d>
    </payment>
```

7.7 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>eJxtk11PgAUhu</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>
```

7.8 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>
```

8 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.

8.1 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.

8.2 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 6.4). With the response the merchant will get the current status of the requested transaction.

8.2.1 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 is generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For DÜ only DEBIT is allowed.

8.2.2 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-8485-832291144af</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>288eeaa56-934e-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>rqDIW1dmCaOcmzGatkW7sjc5kA0=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>epUaP8or3Y3ozKb2DgvPHikqhh2E8QTToh8IF2yOe0254oWA+rEEID0
    1TyMOCWw0a5bQNIBUNx3V
    0inZ9DOm5U9xfT6vEPiU1IN+ekUCtnFXnFj/W5F6k8qZ9EqFZ9yk1DKdM+7IwuFfAyg
    52ObHEiRM
    b1sICqaX/RwqU+K3sNeVE3yw4YUyIrso0DNJLLap/w6i4/bjMOdptHqOD+5pTCpDKb7
    mVnLHwpF6
    FFszWCXssCZM99ILWBG6ycXSCZ6Pok8wSuDykbv8fRmXk+PTd/efnDzUFLIdq7x/Op
    1R6qj1plU3 Edbo43BdiuZyRj3KAjkAJco9VGwJpz7UPzoYPA==</SignatureValue>

```

```

- <KeyInfo>
  - <X509Data>
    <X509SubjectName>CN=xxx,OU=Operations,O=Wirecard Elastic Payments,L=xxx,ST=ON,C=CA</X509SubjectName>
    <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6M
      QswCQYDVQQGEwJDQTELMAkGA1UE
      CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsY
      XN0aWMgUGF5bVVu
      dHMxEzARBgNVBAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUw
      HhcNMTEwNjI0MDQ0
      NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UE
      CBMCT04xEDAOBgNVBAcT
      B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bVVuHm
      EzARBgNVBAsTCk9w
      ZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBA
      QUAA4IBDwAwggEK
      AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xBozFZ3Vh3XQyy3IpIuEfZz7004
      k4HeonfTxCN
      etBvJ9rgNc0Cxrk/euMj3pOURE9WYN2eAXC0r5pUIAZhIANsXUSAIF3JKBxf7gDAi
      k5d8RT5HaJV
      4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyH
      IIAMGzW7VgKD
      2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvn
      RWbIzyQuyUyDeUb
      QmhVQKL0WIb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIh
      vcNAQEFBQADggEB
      ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2p
      bOD3HBXoyovZr
      mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89S
      SucBfIS5TJ54e7m
      02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGxfOWz2fKmeZBuTeY+HiTEH8
      KCHpfw2j8G+dDgU
      jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG
      13AzYwDP3
      x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
</payment>

```

8.3 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 6.4). With the response the merchant will get the current status of the requested transaction.

8.3.1 Fields

The following elements are mandatory (X) / optional (O) for sending request/response for the Payment Method **Sofortbanking**.

Field	Request	Response	Notification	Data Type	Size	Description
merchant-account-id	X	X	X	String	36	Unique identifier for a merchant account.
transaction-id		X	X	String	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	X	X	X	String	150	This is the identification number of the request. It has to be unique for each request.
transaction-type	X	X	X	String	30	This is the type for a transaction. For Skrill Wallet only DEBIT is allowed in the initial request.
transaction-state		X	X	String	12	This is the status of a transaction
completion-time-stamp		X	X	dateTime		This is the timestamp of completion of request.
status.code		X	X	String	12	This is the code of the status of a transaction.
status.description		X	X	String	256	This is the description to the status code of a transaction.
status.severity		X	X	String	20	This field gives information if a status is a warning, an error or an information.
statuses.Status		X	X	String	12	This is the status of a transaction.
requested-amount	X	X	X	Decimal	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
parent-transaction-id	O	O	O	String	36	Transaction ID of the first transaction of a payment.
account-holder.address.city	O	O	O	String	32	This is the end-consumer's city.
account-holder.address.Country	O	O	O	String	3	This is the end-consumer's country.
account-holder.address.postal-code	O	O	O	String	16	This is the end-consumer's postal code.
account-holder.address.state	O	O	O	String	32	This is the end-consumer's state.

account-holder.address.street1	O	O	O	String	128	This is the first part of the end-consumer's street.
account-holder.address.street2	O	O	O	String	128	This is the second part of the end-consumer's street.
account-holder.date-of-birth	O	O	O	Date	0	This is the end-consumer's birth date.
account-holder.email	O/X	O/X	O/X	String	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	O	O	O	String	27	This is the first name of the end-consumer. The maximum size of first-name and last-name in combination is 27 characters.
account-holder.gender	O	O	O	String	1	This is the end-consumer's gender.
account-holder.last-Name	O	O	O	String	27	This is the last name of the end-consumer. The maximum size of first-name and last-name in combination is 27 characters.
account-holder.phone	O	O	O	String	32	This is the phone number of the end-consumer.
bank-account.account-number	O/X	O/X	O/X	String	32	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/X	O/X	O/X	String	32	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.iban	X/O	X/O	X/O	String	32	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 are the following characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})?)
bank-account.bic	X/O	X/O	X/O	String	32	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 are the following characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})?)
ip-address	O	O	O	String	15	The global (internet) IP address of the consumers computer.
order-number	X	X	X	String	64	This is the order number of the merchant.
order-Detail	O	O	O	String	65535	This is a field for details of an order filled by the merchant.
descriptor	X	X	X	String	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', ' ', '+', ',', '-' , ''
notifications.notification@url	O	O	O	String	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
custom-field.field-name	O	O	O	String	36	This is the name for the custom field.
custom-field.field-value	O	O	O	String	256	This is the content of the custom field. In this field the merchant can send additional information.
payment-methods.payment-method@name	X	X	X	String	15	This is the name of the payment method that

						that is chosen from the end-consumer. Value "sofortbanking" should be used.
api-id			X	String		The API id is always returned in the notification. For Skrill it is “---”
processing-redirect-URL	O	O	O	String	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.
cancel-redirect-URL	X	X	X	String	256	The URL to which the consumer will be redirected after he has cancelled a payment. This is normally a page on the merchant's Website.
success-redirect-URL	X	X	X	String	256	The URL to which the consumer will be redirected after a successful payment. This is normally a success confirmation page on the merchant's website.
Signature			X	String		The Signature info, consisting of SignedInfo, SignatureValue and KeyInfo.

8.3.2 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:

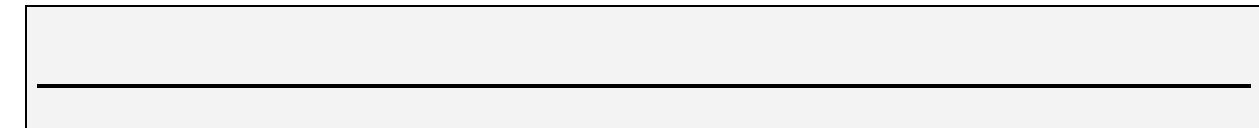
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>96b016e5-ceb2-4c51-a267-0297574f7d25</merchant-
account-id>
    <transaction-id>01c9e586-f1d6-11e1-b0a2-f04da2253560</transaction-id>
    <request-id>2dd77e58-57c7-4363-9679-7073814cb81a</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>44443332221111</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>44443332221111</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><CanonicalizationMet
  hod
    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>nCQV1V6h
  mNKZF4aaej1i6Hsshgc=</DigestValue></Reference></SignedInfo><SignatureValu
  e>bOe6398GUPmi/VwAWfcB/MSTXxp4jYBGhyYZbW1s9gqmKwiDvNNGOXC8S7mLrax
  KifixGmX0S8KQ
  gKH585Jcwne5+GWbLPDK3W/MZ3WaAGn7wr3jylbNae2H2gpZ5x3nAA/0bqjFafF85b5j
  nM8V49uB
  TIWZz04t9A2pTQiP3Mzx4qan37rwym0hVUwGGziscVU6N5EfNf/xqEqH9Zr7Y55CnAB
  Q6dMuTR
  f49s92XIXEcAoIOTnUnSvdhPi1PEx83MmTgOZcK7hDtH1T3LdRm+DNh1Lk0IdKUjhWiG
  LVpzs1ah
  tVBgcHMyNReJFE+9XsvEJfyQabhuDvxPEQLdw==</SignatureValue><KeyInfo><X5
  09Data><X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic
  Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDC
  CAIlgAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAk
  GA1UE
  CBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpdcmVjYXJkIEVsYXN0aW
  MgUGF5bWv
  dHMxEzARBgNVBAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNM
  TEwNjI0MDQ0
  NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT
  04xEDAOBgNVBAcT
  B1Rvcn9udG8xIjAgBgNVBAoTGVdpdcmVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBg
  NVBAsTCk9w
```

```

ZXJhdGlvbMxEzARBgNVBAMTck1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA
4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzIsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4Heon
fTxCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIANsxUsaIF3JKBxf7gDAik5d8RT
5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIIAMGz
W7VgKD
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvnrWbIz
yQuyUyDeUb
QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAEwDQYJKoZIhvcNAQ
EFBQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3H
BXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfI
S5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpf
w2j8G+dDgU
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFoJNdG13AzY
wDP3
x/QspK0jYn1KZw1qz524VWoQouR8Xj30A2jntA=</X509Certificate></X509Data><
KeyInfo></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><CanonicalizationMet
  hod
  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>PZLE514w
  GOwOhK5tfyAKeGfyg0c=</DigestValue></Reference></SignedInfo><SignatureValu

```

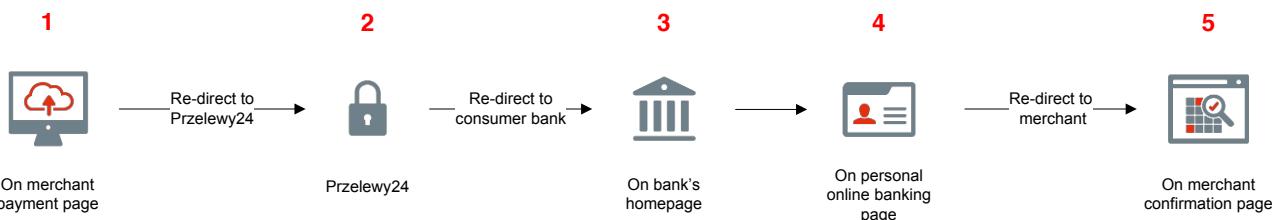
```
e>TY2i/KdIXRW2xlhK/fRmY9SIXNB+xG+EILmDecaUuUOrGENYHZZCgdQTY2BKftuMM  
nu7h4B6eREw  
vfmhwX79kP386DUrsxJjypkAbr/Q5w35eVdrOXGsKyBN3FZWXmfOAgXghZa0y4MXKPM  
qYweUHakd  
9+y+ToQsvjuKTuoxdgOyZ0d6dUMBI1v0GG3o1YaGE1WYRaZjZdUA6mvVzXzg23QG9Q  
Yt0pt5koUz  
Zvkx1k2hEdonWQ2CebwOfLWWK/fmXmZDiCuxhQiU4R05Y9yABpR6OqkfN9wCmV928  
atNV8sTPyXQ  
5ttJshdWqxQkNDHwB8YLBC EUY3OTBQB sLbZCjw==</SignatureValue><KeyInfo><X  
509Data><X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic  
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDC  
CAli gAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMak  
GA1UE  
CBMCT04xEDAOBgNVBAcTB1Rvc m9udG8xIjAgBgNVBAoTGVdp cmVjYXJkIEVsYXN0aW  
MgUGF5bWVu  
dHMxEzARBgNVBAsTCk9wZXJhdGl vbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNM  
TEwNjI0MDQ0  
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMakGA1UECBMCT  
04xEDAOBgNVBAcT  
B1Rvc m9udG8xIjAgBgNVBAoTGVdp cmVjYXJkIEVsYXN0aWMgUGF5bWVu dHMxEzARBg  
NVBAsTCk9w  
ZXJhdGl vbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIB3DQEBAQUAA  
4IBDwAwggEK  
AoIBAQ Cc8rTt4N5fNeVzlsRg0XKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4Heon  
fTxCN  
etBvJ9rgNc0Cxrk/euMj3pOUR E9WYN2eAXC0r5pUIAZhIAN SxUSA IF3JKBxf7gDAik5d8RT  
5HaJV  
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHIIAMGz  
W7VgKD  
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkv nRWbIz  
yQuyUyDeUb  
QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LA gMBAAEwDQYJKoZIhvcNAQ  
EFBQADggEB  
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3H  
BXoyovZr  
mk/KqOyUL+unVcR+APjxX4KP25sd kplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfI  
S5TJ54e7m  
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpf  
w2j8G+dDgU  
jlp9Lv jVN mJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NF foJNdG13AzY  
wDP3  
x/QspK0jYn1KZw1qz524VWoQou eR8Xj30A2jntA=</X509Certificate></X509Data></  
KeyInfo></Signature></payment>
```

8.4 Przelewy24

Przelewy24 is a payment method offering online bank transfers to over 30 banks in Poland.

The end consumer is redirected to Przelewy24 and authorizes the transaction. A notification confirming the outcome of the transaction is sent to the merchant.

The notification for the final status of a payment should be received within a few minutes for a “Przelewy24” 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 6.4). With the response the merchant will get the current status of the requested transaction.



8.4.1 Fields

The following elements are mandatory/optional for sending a request/response for the payment method Przelewy24

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
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
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 is generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For DÜ only DEBIT is allowed.
payment-methods/payment-method/@url	M	Alphanumeric	2056	This is the URL which the consumer must be redirected to in order to complete the payment. *Note: for Przelewy24, the parameters must be URL decoded and posted to the url portion of the payment-method@url.

8.4.2 XML Samples for Przelewy24 Debit

Sample of a successful Przelewy24 Debit Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <request-id>${=(int)(Math.random()*1000000)}</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="PLN">1.01</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="p24" />
  </payment-methods>
  <notifications>
    <notification url="https://merchant.com/ipn.php" />
  </notifications>
  <success-redirect-url>https://merchant.com/success.php</success-redirect-url>
  <cancel-redirect-url>https://merchant.com/cancel.php</cancel-redirect-url>
  <fail-redirect-url>https://merchant.com/fail.php</fail-redirect-url>
</payment>
```

Sample of a successful Przelewy24 Debit Response:

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

```

<merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
<transaction-id>5ba83eba-a463-11e5-8105-005056a96a54</transaction-id>
<request-id>940846</request-id>
<transaction-type>debit</transaction-type>
<transaction-state>success</transaction-state>
<completion-time-stamp>2015-12-17T02:10:35.000Z</completion-time-stamp>
<statuses>
  <status code="201.0000" description="The resource was successfully created." severity="information" />
</statuses>
<requested-amount currency="PLN">1.01</requested-amount>
<account-holder>
  <first-name>John</first-name>
  <last-name>Doe</last-name>
</account-holder>
<notifications>
  <notification url="https://merchant.com/ipn.php"/>
</notifications>
<payment-methods>
  <payment-method url="https://sandbox.przelewy24.pl/index.php?p24_session_id=5ba83eba-a463-11e5-8105-005056a96a54&p24_id_sprzedawcy=16027&p24_kwota=101&p24_klient=John+Doe&p24_return_url_ok=https%3A%2F%2Fapi-test.wirecard.com%3A443%2Fengine%2Fnotification%2Fp24%2F&p24_return_url_error=https%3A%2F%2Fapi-test.wirecard.com%3A443%2Fengine%2Fnotification%2Fp24%2F&p24_language=EN&p24_crc=89b9c4bdacde2cc20a0df0ef43e031d3&" name="p24"/>
</payment-methods>
<cancel-redirect-url>https://merchant.com/cancel.php</cancel-redirect-url>
<fail-redirect-url>https://merchant.com/fail.php</fail-redirect-url>
<success-redirect-url>https://merchant.com/success.php</success-redirect-url>
</payment>

```

Sample of a successful Przelewy24 Debit Notification:

```

<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <transaction-id>a11ce9fc-a461-11e5-8105-005056a96a54</transaction-id>
  <request-id>401623</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-12-17T01:58:13.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information" />
  </statuses>
  <requested-amount currency="PLN">1.010000</requested-amount>
  <parent-transaction-id>837b5dc0-a461-11e5-8105-005056a96a54</parent-transaction-id>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <payment-methods>
    <payment-method name="p24" />
  </payment-methods>
  <api-id>---</api-id>
  <success-redirect-url>https://merchant.com/success.php</success-redirect-url>
  <cancel-redirect-url>https://merchant.com/cancel.php</cancel-redirect-url>
  <fail-redirect-url>https://merchant.com/fail.php</fail-redirect-url>
  <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>viFFrbhG6DZGQXI6wo/FGL3KbBk=</DigestValue>
  </Reference>
</SignedInfo>

<SignatureValue>OpZyTylwjUJrnix1No4eRwg3+lotAskH6LFuosVsf2COFRQWF2Q3u9iT5+lqqbaUyYGkRBpa6FRmeE
SVsKa6H4H+Pm/RSAzzOgtFLeX7BtkB3JQBCnGaUJBtKoo9Pl6mfNrqvHdO9vviQPtNZ4NfMx5W/Ekv1wvgnCdOTURXqf
4MW20EiLEAV0h2j3jB6gNlxRdBuyOg5WZHA62qz8IRV0BPbNMqnyRsMuArhyziMjeZT6Txh5IAFRY2sOcGu8aw6Bk
mnYS6Fi0nH7jkgt9psNEkSqw8jYMGE1C0L3bIXWSHYI1DuYL+hycB42RBT6+5piOlqWY186ofi+20FVg==</SignatureVa
lue>
  <KeyInfo>
    <X509Data>
      <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>

      <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkG
A1UECBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpdmVjYXJkIEVsYXN0aWMgUGF5bVVudHMxEzA
RBgNVBAAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qlFNhaHUwHhcNMTEwNjI0MDQ0NDA5WhcNMTQwMzl
wMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBA
oTGVdpdmVjYXJkIEVsYXN0aWMgUGF5bVVudHMxEzARBgNVBAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9ql
FNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xB0zFZ3Vh3
XQyy3lpluEfZz7004k4HeonTxCNetBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAnSxUSAIf3JKBxf7gDAik5d8
RT5HaJV4n5cXJQ/uhAEYU3EGN/74UrD2UsODY3VBXTJS5VgSi/c3lyLwhDbYIyU6j4fMKyHIIAMGzW7VgKD2pqu6BRys
qUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvnRWbIzyQuyUyDeUbQmhVQKL0WIMb5ev65m2
VjGyDTGL5jfB14rSXRMGzeJ+LAGMBAEwDQYJKoZlhvcNAQEFBQADggEBADgkuN/e2Ify7JXdbjNjbKBd3HLvFvK87dv8
qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZrmk/KqOyUL+unVcR+APjxX4KP25sdplgmeQ47CWxtKAHZUTtWw
AVI/WhsX89SSucBflS5TJ54e7m02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHp
fw2j8G+dDgUlp9LvjVNmjzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFFoJNdG13AzYwDP3x/QspK0jYn
1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
</payment>

```

Sample of a failed Przelewy24 Debit Request:

```

<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <request-id>999513</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="PLN">1.01</requested-amount>
  <payment-methods>
    <payment-method name="p24" />
  </payment-methods>
  <notifications>
    <notification url="https://merchant.com/ipn.php" />
  </notifications>
  <success-redirect-url>https://merchant.com/success.php</success-redirect-url>

```

```
<cancel-redirect-url>https://merchant.com/cancel.php</cancel-redirect-url>
<fail-redirect-url>https://merchant.com/fail.php</fail-redirect-url>
</payment>
```

Sample of a failed Przelewy24 Debit Response:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
<transaction-id>f3bf627e-a462-11e5-8105-005056a96a54</transaction-id>
<request-id>999513</request-id>
<transaction-type>debit</transaction-type>
<transaction-state>failed</transaction-state>
<completion-time-stamp>2015-12-17T02:07:40.000Z</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="PLN">1.01</requested-amount>
<notifications>
  <notification url="https://requestb.in/pf1r2qpf"/>
</notifications>
<payment-methods>
  <payment-method name="p24"/>
</payment-methods>
<success-redirect-url>https://merchant.com/success.php</success-redirect-url>
<cancel-redirect-url>https://merchant.com/cancel.php</cancel-redirect-url>
<fail-redirect-url>https://merchant.com/fail.php</fail-redirect-url>
</payment>
```

8.4.3 XML- Samples for Przelewy24 Refund Debit

URL	https://api-test.wirecard.com/engine/rest/payments
Request Formats	XML
Response Formats	XML
Request Methods	POST
Transaction Type	refund-debit

Sample of a successful Przelewy24 Refund Debit Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
<request-id>c72141d3-0df3-4dd3-9847-dac2198b790f</request-id>
<transaction-type>refund-debit</transaction-type>
<parent-transaction-id>1844d1a9-a4cb-11e5-9970-00163e5933ee</parent-transaction-id>
<ip-address>127.0.0.1</ip-address>
<payment-methods>
  <payment-method name="p24"/>
</payment-methods>
</payment>
```

Sample of a successful Przelewy24 Refund Debit Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment" self="https://api-test.wirecard.com/engine/rest/merchants/5f15a3cf-5cea-4dac-873c-2e5e46225cdf/payments/26a21036-a4cb-11e5-9970-00163e5933ee">
  <merchant-account-id ref="https://api-test.wirecard.com/engine/rest/merchants/5f15a3cf-5cea-4dac-873c-2e5e46225cdf">5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <transaction-id>26a21036-a4cb-11e5-9970-00163e5933ee</transaction-id>
  <request-id>c72141d3-0df3-4dd3-9847-dac2198b790f</request-id>
  <transaction-type>refund-debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-12-17T14:33:34.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="p24:The resource was successfully created." severity="information" provider-transaction-id="597377" />
  </statuses>
  <requested-amount currency="PLN">10.00</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email>john.doe@wirecard.com</email>
    <phone>5555555555</phone>
    <address>
      <street1>150 Eglinton</street1>
      <street2>Suite 999</street2>
      <city>Toronto</city>
      <state>ON</state>
      <country>CA</country>
      <postal-code>M4P1E8</postal-code>
    </address>
  </account-holder>
  <ip-address>127.0.0.1</ip-address>
  <payment-methods>
    <payment-method name="p24" />
  </payment-methods>
  <api-id>elastic-api</api-id>
</payment>
```

Sample of a failed Przelewy24 Refund Debit Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <request-id>15778</request-id>
  <transaction-type>refund-debit</transaction-type>
  <parent-transaction-id>d0efabc8-a462-11e5-8105-005056a96a54</parent-transaction-id>
</payment>
```

Sample of a failed Przelewy24 Refund Debit Response:

```
<payment self="https://api-test.wirecard.com:443/engine/merchants/5f15a3cf-5cea-4dac-873c-2e5e46225cdf/payments/76f1eee4-a465-11e5-8105-005056a96a54" xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id ref="https://api-test.wirecard.com:443/engine/merchants/5f15a3cf-5cea-4dac-873c-2e5e46225cdf">5f15a3cf-5cea-4dac-873c-2e5e46225cdf</merchant-account-id>
  <transaction-id>76f1eee4-a465-11e5-8105-005056a96a54</transaction-id>
```

```

<request-id>15778</request-id>
<transaction-type>refund-debit</transaction-type>
<transaction-state>failed</transaction-state>
<completion-time-stamp>2015-12-17T02:25:40.000Z</completion-time-stamp>
<statuses>
  <status code="500.1087" description="The connection to the provider was never made and timed out.
Please try again later." severity="error"/>
</statuses>
<requested-amount currency="PLN">1.01</requested-amount>
<account-holder>
  <first-name>John</first-name>
  <last-name>Doe</last-name>
</account-holder>
<payment-methods>
  <payment-method name="p24"/>
</payment-methods>
<api-id>elastic-api</api-id>
<cancel-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp</cancel-redirect-url>
<fail-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp</fail-redirect-url>
<success-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp</success-redirect-url>
</payment>
```

9 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.

9.1 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/>.

9.2 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 6.4).

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 appear 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 preferred payment method). This debit-return will be sent from Wirecard to the merchant within an additional notification.

9.2.1 Fields

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

Field	Request	Response	Notification	Data Type	Size	Description
merchant-account-id	X	X	X	String	36	Unique identifier for a merchant account.
transaction-id		X	X	String	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	X	X	X	String	150	This is the identification number of the request. It has to be unique for each request.
transaction-type	X	X	X	String	30	This is the type for a transaction. For Skrill Wallet only DEBIT is allowed in the initial request.
transaction-state		X	X	String	12	This is the status of a transaction
completion-time-stamp		X	X	dateTime		This is the timestamp of completion of request.
status.code		X	X	String	12	This is the code of the status of a transaction.

status.description		X	X	String	256	This is the description to the status code of a transaction.
status.severity		X	X	String	20	This field gives information if a status is a warning, an error or an information.
statuses.Status		X	X	String	12	This is the status of a transaction.
requested-amount	X	X	X	Decimal	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
parent-transaction-id	O	O	O	String	36	Transaction ID of the first transaction of a payment.
account-holder.address.city	O	O	O	String	32	This is the end-consumer's city.
account-holder.address.Country	O	O	O	String	3	This is the end-consumer's country.
account-holder.address.postal-code	O	O	O	String	16	This is the end-consumer's postal code.
account-holder.address.state	O	O	O	String	32	This is the end-consumer's state.
account-holder.address.street1	O	O	O	String	128	This is the first part of the end-consumer's street.
account-holder.address.street2	O	O	O	String	128	This is the second part of the end-consumer's street.
account-holder.date-of-birth	O	O	O	Date	0	This is the end-consumer's birth date.
account-holder.email	X/O	X/O	X/O	String	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	X	X	X	String	27	This is the first name of the end-consumer. The maximum size of first-name and last-name in combination is 27 characters.
account-holder.gender	O	O	O	String	1	This is the end-consumer's gender.
account-holder.last-Name	X	X	X	String	27	This is the last name of the end-consumer. The maximum size of first-name and last-name in combination is 27 characters.

account-holder.phone	O	O	O	String	32	This is the phone number of the end-consumer.
ip-address	O	O	O	String	15	The global (internet) IP address of the consumers computer.
order-number	X	X	X	String	64	This is the order number of the merchant.
order-Detail	O	O	O	String	65535	This is a field for details of an order filled by the merchant.
descriptor	X	X	X	String	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','','+',',','.'
notifications.notification@url	O	O	O	String	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
custom-field.field-name	O	O	O	String	36	This is the name for the custom field.
custom-field.field-value	O	O	O	String	256	This is the content of the custom field. In this field the merchant can send additional information.
payment-methods.payment-method-Name	X	X	X	String	15	This is the name of the payment method that that is chosen from the end-consumer.
api-id			X	String		The API id is always returned in the notification. For Skrill it is "___"
processing-redirect-URL	O	O	O	String	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.
cancel-redirect-URL	X	X	X	String	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.

success-redirect-URL	X	X	X	String	256	The URL to which the consumer will be redirected after a successful payment. This is normally a success confirmation page on the merchant's website.
locale	X	X	X	String	6	Code of the language. Can be any of CZ, DA, EN, DE, ES, FI, FR, IT, NL, PL, GR, RO, RU, SV and TR. Can be sent in the format <language> or in the format <language_country>.
wallet-account-id	X/O	X/O	X/O	String	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 initial request.
settlement.gross-amount	X	X	X	decimal	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.currency	X	X	X	String	3	The currency in which the Merchant Account can expect his Payment Settlement Amount.
Signature			X	String		The Signature info, consisting of SignedInfo, SignatureValue and KeyInfo

9.2.2 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-ae62cdffadfd1</request-id>
<transaction-type>debit</transaction-type>
<requested-amount currency="EUR">10.01</requested-amount>
```

```

<account-holder>
    <!-- either set mandatory 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 mandatory 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-ae62cdffadef1</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>MÃ¼nchen</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+sdezj7ZUttkLOQxxcD4NmnnUcVtFtDnZc73vjWSTL6pBsfl9vXMGh
qoeAzaGpTlbEKQ5rm4Z7oKfWnjPiO4rQ4Cj4uUli8EqCBvqCOE6NRFOkFNVWBxYOLx0wEphVxf
+xTQ3oJiw+8tbmbg7U9fUVKTWlmyJ2jMOf+1gaKxWCBS7pUTX5qAK/o7rRgiO5DBFOhql2eqyOk
B8V0rsYUBLflSJdeOlh4iQ1DTNhXxr5VVNEdI1Fknda+APut6gpQ3rWM+UOYypXMIMaLj2UPMTv3
PjeL7C+dNVJ+15qRfc0QL+9l/oVL5iZfO3tFKw==</SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>
      <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA
1UE
CBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpdmVjYXJkIEVsYXN0aWMgUGF5bWVu
dHMxEzARBgNVBAAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qlFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzlwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcT
B1Rvcn9udG8xIjAgBgNVBAoTGVdpdmVjYXJkIEVsYXN0aWMgUGF5bWVuHdHxEzARBgNVBAAsTCk9w
ZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qlFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
```

```
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xBozFZ3Vh3XQyy3IpluEfZz7004k4HeonfTxCN  
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAAnSxUSaIF3JKBxf7gDAik5d8RT5HaJV  
4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3lyLwhDbYlyU6j4fMKyHIIAMGzW7VgKD  
2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSar3CvYYxYqBN2KBUjabkvnRWblzyQuyUyDeUb  
QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAGMBAAEwDQYJKoZlhvcNAQEFBQADggEB  
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr  
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBflS5TJ54e7m  
02qvGoK8UA/IrbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU  
jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRKO/sJtZBzQajpnsZ9NFfoJNdG13AzYwDP3  
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>  
    </X509Data>  
  </KeyInfo>  
  </Signature>  
</payment>
```

9.3 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.

9.3.1 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.

9.3.2 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>02fce8ce-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>02fce8ce-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>
```

9.4 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 9.4.3 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.

9.4.1 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

9.4.2 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.

From mid-2015 we added a Provider Transaction ID into each response. Provider Transaction ID is returned in response of a **DoExpressCheckoutPayment**, **DoCapture**, **RefundTransaction** and **DoReferenceTransaction** method and is also in a notification for the following transaction types: Authorization, Refund, Capture and Debit. This response and notification is a confirmation that a transaction has been really done.

The provider transaction ID can be found in <Status> field and it is generated for every transaction by PayPal. It works for every merchant and provides an additional information for merchant. The transaction id is stored in a DB in ProviderTransactions table.

```
<?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>
```

```
</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="201.000" description="The resource was successfully
created." severity="information" provider-transaction-id="NMTPYE7FQWYFRGQDJ"/>
    </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&useraction=commit&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" provider-transaction-id="NMTPYE7FQWYFRGQDJ"/>
    </statuses>
    <requested-amount currency="EUR">1.03000</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+Ppr5ZgyUNuJsjq8Ry76F3YM2iEcVYNMd5FM24pXwSLPF0SWuush/XjoXw+n
zLSEhCe8S9dYbQ
peC5Y7/7xoYIYPi2Pwp85JD4bZmKPB+K5GVGo2hMsjMICWDdsQ1e3h2Kf5wgND+puGwCNCe5VRb4
BTZOMhn/5QiSp9VAeBGG1AoLj10302JF6ykt1kZdB1RJX6DGyXNOrUjSHKUhSSj7+6ZJrNot28+E
YT1/RUX4CVD2R724zFMm4NrgNML7znNUkdDA/e4SwBbXJ2gKX8LskII9EeKTxahi2su3G+HGg1
YSGjTwt+EiyhdKj9e2LHbw7vjP0gfBRV6bf36w==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard
Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>
                <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGE
wJDQTELMAkGA1UE
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWlgUGF5bwVu
dHMxEzARBgNVBAstCk9wZXJhdGlvbnMxEzARBgNVBAMTck1hbmq9qIFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWlgUGF5bwVuHdMxEzARBgNVBAstCk9w
ZXJhdGlvbnMxEzARBgNVBAMTck1hbmq9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUfJx7xXBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCN
etBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIAhSxUsaIF3JKBxf7gDAik5d8RT5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2Us0YD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyHI1AMGzW7VgKD
2pqu6BRysqUVdEEAvW20myVqGVyPk87EiHSMMSSar3CvYYxYqBN2KBUjabkvnrWbIzyQuyUyDeUb
QmhVQKL0W1Mb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvCNQEFBQADggEB
ADgkuN/e2IFy7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHSA6C2pbOD3HBXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CwxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXF0Wz2fKmeZBuTeY+HiTEH8KChpfw2j8G+dDgU
jlp9LvjVNmjzfNBBK1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
            </X509Data>
        </KeyInfo>
    </Signature>
</payment>
```

A sample of a failed request and response. Please note that there is no notification sent back to merchant.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>c57f418f-81ff-44dc-8acf-df9524c080df</merchant-account-id>
    <request-id>01fb2808-7ad5-41eb-b883-b6b522f18bac</request-id>
    <transaction-type>authorization</transaction-type>
    <payment-methods>
        <payment-method name="paypal" />
    </payment-methods>
    <requested-amount currency="EUR">-10.01</requested-amount>
    <order-description>test order</order-description>
    <descriptor>customerStatement 18009998888</descriptor>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
    </account-holder>
    <order-number>4469</order-number>
    <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>
</payment>
```

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>c57f418f-81ff-44dc-8acf-df9524c080df</merchant-account-id>
    <transaction-id>3951370d-15cc-11e5-b217-ecf4bb4519f3</transaction-id>
    <request-id>01fb2808-7ad5-41eb-b883-b6b522f18bac</request-id>
    <transaction-type>authorization</transaction-type>
    <transaction-state>failed</transaction-state>
    <completion-time-stamp>2015-06-18T15:11:56.000Z</completion-time-stamp>
    <statuses>
        <status code="400.1013" description="The Requested Amount is below the minimum required for this Merchant Account. 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>
    <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>4469</order-number>
    <descriptor>customerStatement 18009998888</descriptor>
```

```

<payment-methods>
    <payment-method name="paypal" />
</payment-methods>
</payment>

```

9.4.3 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.

9.4.4 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.

Request, the first payment

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>495d7083-3dc9-444d-bac6-3e110f2c8609</merchant-account-id>
  <request-id>b4ea36bb-20b9-4fc7-9386-4ffa74d30db0</request-id>
  <transaction-type>authorization</transaction-type>
  <api-id>elastic-payment-page</api-id>
  <payment-methods>
    <payment-method name="paypal"/>
  </payment-methods>
  <requested-amount currency="EUR">1.03</requested-amount>
  <order-description>test order</order-description>
  <descriptor>customerStatement 18009998888</descriptor>
  <account-holder>
    <first-name>SandboxTest</first-name>
    <last-name>Account</last-name>
  </account-holder>
  <order-number>40795</order-number>
  <shipping>
    <first-name>SandboxTest</first-name>
    <last-name>Account</last-name>
    <phone>+49123123123</phone>
    <address>
      <street1>ESpachstr. 1</street1>
      <city>Freiburg</city>
      <country>DE</country>
      <postal-code>79111</postal-code>
    </address>
  </shipping>
  <periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>first</sequence-type>
  </periodic>
</payment>
```

Response

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>495d7083-3dc9-444d-bac6-3e110f2c8609</merchant-account-id>
  <transaction-id>0692804a-57b3-11e5-a5b6-005056a9092e</transaction-id>
  <request-id>b4ea36bb-20b9-4fc7-9386-4ffa74d30db0</request-id>
  <transaction-type>authorization</transaction-type>
  <transaction-state>success</transaction-state>
```

```

<completion-time-stamp>2015-09-10T11:56:53.000Z</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>SandboxTest</first-name>
    <last-name>Account</last-name>
</account-holder>
<shipping>
    <first-name>SandboxTest</first-name>
    <last-name>Account</last-name>
    <phone>+49123123123</phone>
    <address>
        <street1>ESpachstr. 1</street1>
        <city>Freiburg</city>
        <country>DE</country>
        <postal-code>79111</postal-code>
    </address>
</shipping>
<order-number>40795</order-number>
<descriptor>customerStatement 18009998888</descriptor>
<payment-methods>
    <payment-method url="https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout&useraction=commit&token=EC-7H1523556G461025D" name="paypal"/>
</payment-methods>
<api-id>elastic-payment-page</api-id>
<periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>first</sequence-type>
</periodic>
</payment>

```

Subsequent referenced capture payment referencing the initial ‘first’ payment based on Parent Transaction Id:

Request, the second payment

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>495d7083-3dc9-444d-bac6-3e110f2c8609</merchant-account-id>
    <request-id>06305afb-45b2-47a0-b1a8-9780580c81e7</request-id>
    <transaction-type>capture-authorization</transaction-type>
    <api-id>elastic-payment-page</api-id>
    <requested-amount currency="EUR">1.03</requested-amount>
    <parent-transaction-id>1914ccb4-57b3-11e5-a5b6-005056a9092e</parent-transaction-id>
    <payment-methods>
        <payment-method name="paypal"/>
    </payment-methods>
</payment>

```

Response, the second payment

```

<payment self="http://q-elastic-tom0.wirecard.sys:9000/engine/rest/merchants/495d7083-3dc9-444d-bac6-3e110f2c8609/payments/1fcad01c-57b3-11e5-a5b6-005056a9092e"
    xmlns="http://www.elastic-payments.com/schema/payment">

```

```

<merchant-account-id ref="http://q-elastic-
tom0.wirecard.sys:9000/engine/rest/merchants/495d7083-3dc9-444d-bac6-
3e110f2c8609">495d7083-3dc9-444d-bac6-3e110f2c8609</merchant-account-id>
<transaction-id>1fcad01c-57b3-11e5-a5b6-005056a9092e</transaction-id>
<request-id>06305afb-45b2-47a0-b1a8-9780580c81e7</request-id>
<transaction-type>capture-authorization</transaction-type>
<transaction-state>success</transaction-state>
<completion-time-stamp>2015-09-10T11:57:36.000Z</completion-time-stamp>
<statuses>
  <status code="201.0000" description="paypal:The resource was successfully created." severity="information" provider-transaction-id="2LD80963TT927794J"/>
</statuses>
<requested-amount currency="EUR">1.03</requested-amount>
<account-holder>
  <first-name>SandboxTest</first-name>
  <last-name>Account</last-name>
  <email>buyer2@wirecard.com</email>
</account-holder>
<shipping>
  <first-name>SandboxTest</first-name>
  <last-name>Account</last-name>
  <address>
    <street1>ESpachstr. 1</street1>
    <city>Freiburg</city>
    <state>Empty</state>
    <country>DE</country>
    <postal-code>79111</postal-code>
  </address>
</shipping>
<order-number>40795</order-number>
<descriptor>customerStatement 18009998888</descriptor>
<payment-methods>
  <payment-method name="paypal"/>
</payment-methods>
<api-id>elastic-api</api-id>
<periodic>
  <periodic-type>recurring</periodic-type>
</periodic>
<wallet>
  <account-id>79FXFKSN2TV92</account-id>
</wallet>
<instrument-country>DE</instrument-country>
</payment>
```

9.4.5 Pre-authorization

It is possible to obtain consumer's preliminary payment confirmation (along with shipping details from PayPal for example) without really reserving funds on payer's PayPal account. This can be achieved by "preauthorization" transaction, which have the same parameters as "authorization" or "debit" request. Similarly, the Forward-URL to the PayPal Express Checkout is delivered in response, where consumer needs to confirm the payment. In order to finalize the payment, regular "authorization" or "debit" transaction needs to be submitted, referencing the prior "preauthorization" transaction (in "parent-transaction-id" field). The pre-authorization does not need to be cancelled, if merchant decides not to follow up this order attempt.

9.4.6 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 **9.4.4 Reference transactions for PayPal**.

9.4.7 Basket items for PayPal

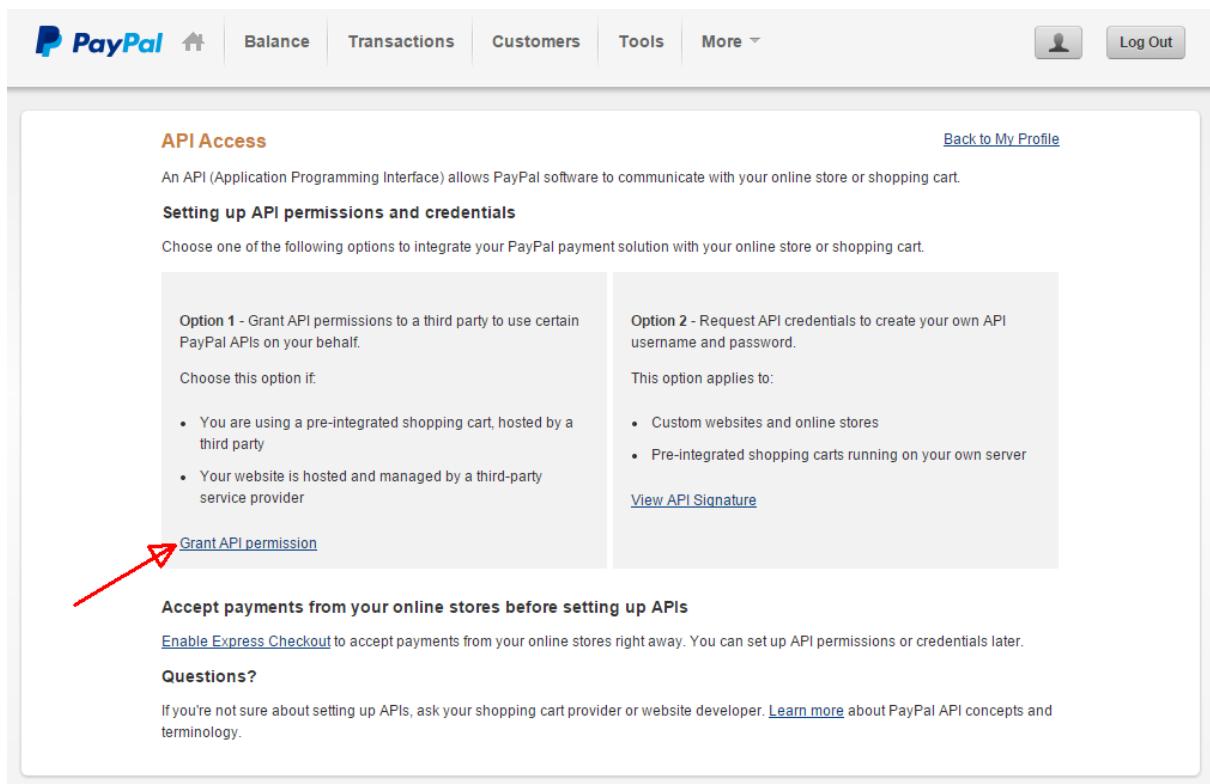
Basket items details might be sent along with debit and authorization requests. The detailed items of the shopping basket will be displayed in PayPal during the checkout, and later stored as a part of payment details in PayPal. They can be obtained from Wirecard Processing API later as part of transaction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

9.4.8 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

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](#)

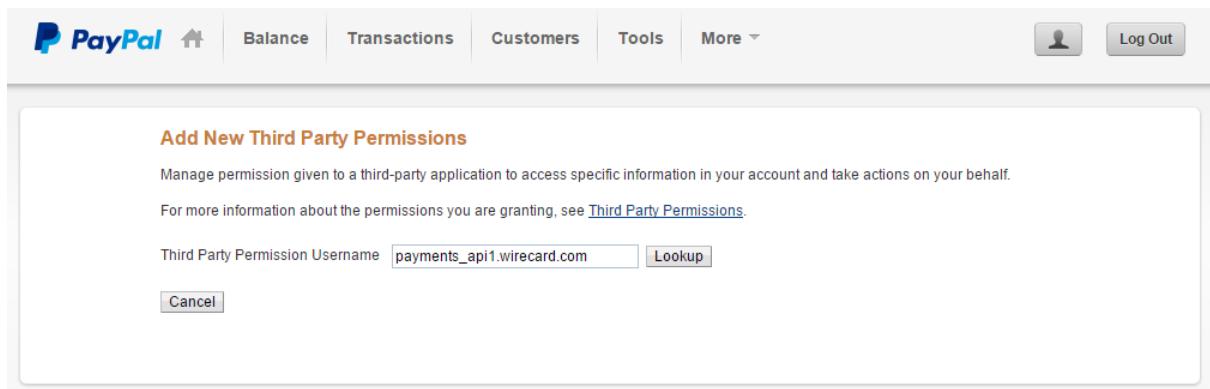
Accept payments from your online stores before setting up APIs

[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 [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

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.

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

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

9.5 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 6.4).

9.5.1 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 redirected after a successful payment.

				This is normally a success confirmation page on the merchant's website.
cancel-redirect-url	0	Alphanumeric	256	The URL to which the consumer will be redirected after he has cancelled a payment. This is normally a page on the merchant's website

9.5.2 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-171ccb86bcf</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-171ccb86bcf</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&URLsuccess=http%3A%2F%2Flocalhost%3A8
080%2Fengine%2Fnotification%2Fekonto&amount=1.01&creditaccount=1212121001&
creditbank=5500&credittext=&curcode=CZK&otherparams=payment.transaction-
id%3D57d70b64-be4b-11e3-be85-0050b66a327a&shopname=EKONTO&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>iZBfp82lilnWdJNnoNxQcqarCI=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>A15px4q0ae6Fp3DZm+9SHv6wOpeS4AIC01+LmCe19JdwmquxeGrh+s6zxz7aewmwzTv05KSHrbYdOV7K8fKINGYLZv/JNvh+URUcvtxoBnHwtp5aKDn8I8TgKNp0y3kkewn5GcknavCie9h10B5YQ50qSWVS+Z9Fm39tD3j1Jg9b1G5xhWlRkF5YODoKr83AQFyL11DVWCWrZzVZEGtsmh01rKgyxSc1XzhSKLhvYYvXUfw9X109GaLt7zfP0Xw5WFuDAd300408hTeExL2Hp5qi9H+0ztlsJSi0+McJ6+vmxmDznT+cXFDG6QvxALJ9KMc28K5K+iXVBe3pA==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>
                <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xE DAOBgNVBAcTB1Rvcm9udG8xIjAgBgnVBaotGVdp cmVjYXJkIEVsYXN0aWMgUGF5bwVudHMxEzARBgNVBAstCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hb m9qIFNhaHUwHhcNMTEwNjI0MDQ0NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MqsWCQYDVQQGEwJDQTELMAkGA1UECBMCT04xE DAOBgNVBAcTB1Rvcm9udG8xIjAgBgnVBaotGVdp cmVjYXJkIEVsYXN0aWMgUGF5bwVudHMxEzARBgNVBAstCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hb m9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCc8rTt4N5fNeVzlsRgOKDE2YUfJx7xBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCNetBvJ9rgNc0Cxrk/euMj3p0UrE9WYN2eAXC0r5pUIAz hIA nSxUsaIF3JKBxf7gDAik5d8RT5hajV4n5cXJQ/uhAEYU3EGN/74UrD2Us0YD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyH1
```

```

    AMGzW7VgKD2pqu6BRysqUVdEEAvW20myVqGVyPkm87EiHSMSar3CvYYxYqBN2KBUjabk
    vnRwbIzyQuyUyDeUbQmhVQKL0w1Mb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAEw
    DQYJKoZIhvCNAQEFBQADggEBADgkuN/e2IFy7JXdbjNjbKbd3HLvFvK87dv8qQ+HK4qfC
    xYXh6aYhbKHJSA6C2pbOD3HBXoyovZrmk/KqOyUL+unVcR+APjxX4KP25sdkp1gmeQ47C
    WxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx
    3ubwwHGXF0Wz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgUjl9LvjVNmJzfNBBk1Si0d/r
    hXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NffoJNdG13AzYwDP3x/QspK0jYn1KZw1qz
    524VWoQoueR8Xj30A2jntA=
  </X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
</payment>

```

A failed request and response, note that there is no notification sent back.

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>36b9dd7c-88a5-4cc9-b25c-
  171ccb86bcf</merchant-account-id>
  <request-id>1f17956e-f3fc-4961-9fd8-70ce6f06fba8</request-id>
  <transaction-type>debit</transaction-type>
  <payment-methods>
    <payment-method name="eplatby" />
  </payment-methods>
  <requested-amount currency="EUR">1.03</requested-amount>
  <order-detail>order detail</order-detail>
</payment>

```

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>36b9dd7c-88a5-4cc9-b25c-
  171ccb86bcf</merchant-account-id>
  <transaction-id>58a0cb79-15d3-11e5-b217-ecf4bb4519f3</transaction-
  id>
  <request-id>1f17956e-f3fc-4961-9fd8-70ce6f06fba8</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>failed</transaction-state>
  <completion-time-stamp>2015-06-18T16:02:55.000Z</completion-time-
  stamp>
  <statuses>
    <status code="400.1168" description="The Order
number is required, but not provided. Please check your input and try again."
severity="error" />
  </statuses>
  <requested-amount currency="EUR">1.03</requested-amount>
  <order-detail>order detail</order-detail>
  <payment-methods>
    <payment-method name="eplatby" />
  </payment-methods>
</payment>

```

9.6 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 9.6.3 and 9.6.4 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.

9.6.1 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
<code>merchant-account-id</code>	M	Alphanumeric	36	Unique identifier for a merchant account
<code>request-id</code>	M	Alphanumeric	150	This is the identification number of the request. It has to be unique for each request.
<code>transaction-type</code>	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.
<code>requested-amount</code>	M	Numeric	18,3	This is the amount of the transaction. The amount of the decimal place is dependent of the currency.
<code>payment-methods.payment-method-name</code>	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.

9.6.2 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-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>

```

Failure response and notification. Please note there is no notification to be sent.

Request:

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>815c32ab-486d-4e4c-baf8-
f412e2e4e9b8</merchant-account-id>
    <request-id>a2e64b08-1da4-4f8c-b4a5-249ac4d9dd60</request-id>
    <transaction-type>authorization</transaction-type>
    <payment-methods>
        <payment-method name="ratepay-elv" />
    </payment-methods>
    <requested-amount currency="EUR">80</requested-amount>
    <order-number>45618</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>27-12-1973</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>
</payment>

```

```

                </address>
            </account-holder>
            <bank-account>
                <!-- one of iban or (account-number and bank code)
mandatory -->
                <account-number>4989123456</account-number>
                <bank-code>20320500</bank-code>
                <bank-name>Danske Bank Hamburg</bank-name>
                <iban>DE99203205004989123456</iban>
                <bic>DABADEHHXXX</bic>
                <account-owner>John Doe</account-owner>
            </bank-account>
            <order-items>
                <order-item>
                    <name>Item 1</name>
                    <article-number>123</article-
number>
                    <amount currency="EUR">80</amount>
                    <tax-rate>0.2</tax-rate>
                    <quantity>1</quantity>
                </order-item>
            </order-items>
            <mandate>
                <mandate-id>12345</mandate-id>
            </mandate>
            <creditor-id>12345</creditor-id>
        </payment>
Response:

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>815c32ab-486d-4e4c-baf8-
f412e2e4e9b8</merchant-account-id>
    <transaction-id>c212f6a1-15ca-11e5-b217-ecf4bb4519f3</transaction-
id>
    <request-id>a2e64b08-1da4-4f8c-b4a5-249ac4d9dd61</request-id>
    <transaction-type>authorization</transaction-type>
    <transaction-state>failed</transaction-state>
    <completion-time-stamp>2015-06-18T15:01:26.000Z</completion-time-
stamp>
    <statuses>
        <status code="400.1110" description="Use either
Bank Account and Bank Code or IBAN and BIC. Please check your input and try
again." severity="error" />
    </statuses>
    <requested-amount currency="EUR">80</requested-amount>
    <account-holder>
        <first-name>Hans-Jürgen</first-name>
        <last-name>Wischnewski</last-name>
        <email>john.doe@test.com</email>
        <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>45618</order-number>

```

```

<order-items>
    <order-item>
        <name>Item 1</name>
        <article-number>123</article-
number>
        <amount currency="EUR">80</amount>
        <tax-rate>0.2</tax-rate>
        <quantity>1</quantity>
    </order-item>
</order-items>
<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>
    <iban>DE99203205004989123456</iban>
    <bic>DABADEHHXXX</bic>
</bank-account>
<mandate>
    <mandate-id>12345</mandate-id>
</mandate>
<creditor-id>12345</creditor-id>
</payment>

```

9.6.3 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 delivered 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 exceed 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.

9.6.4 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.

9.7 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:

- MasterPass debit and credit card purchase
- MasterPass authorization and credit card authorization
- MasterPass authorization-only and credit card authorization-only

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.

URL	https://api-test.wirecard.com/engine/rest/paymentmethods/
TEST Merchant Account ID	8bc8ed6d-81a8-43be-bd7b-75b008f89fa6
TEST Merchant Account Name	WireCard MasterPass
Username	70000-APITEST-AP
Password	qD2wzQ_hrc!8

9.7.1 Merchant configuration / Registration Procedure

9.7.1.1 Step 1 – Initial Registration

Merchant registration is done on the MasterPass Merchant Portal (<https://masterpass.com/SP/Merchant/Home>).

From the MasterPass Merchant Portal, select the country – language from the dropdown and click the **Create an Account** button to start the registration process. You will be presented with a modal window, into which you will enter the invitation code. After entering the invitation code, you will be presented with the option to select the registration type. Select Merchant to continue with the registration process as shown in screen shots below.



Masterpass issued a set of invitation codes to Wirecard. Feel free to use one of these codes. Some of them may have been already used. If we run out of codes, please ask Wirecard Support to obtain more codes.

At this point, "Merchant" needs to be chosen.

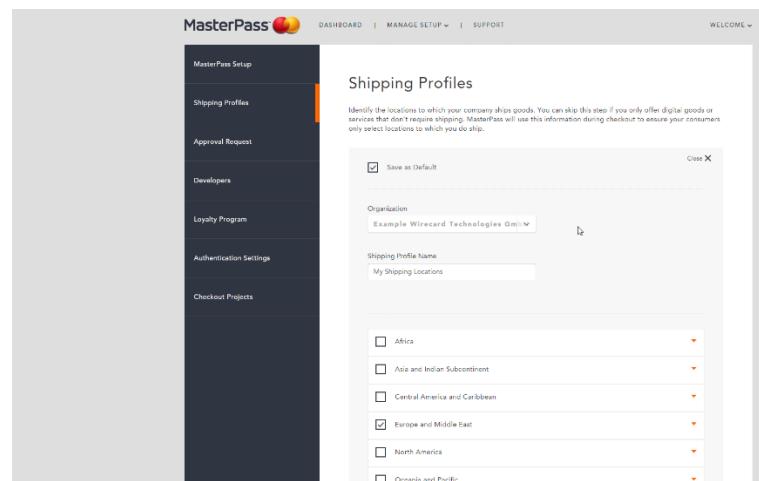
Complete MasterPass registration procedure by filling the required information.

After successful registration, you are logged in to Masterpass Merchant Portal.

The image shows three sequential steps in the MasterPass registration process:

- Invitation Code:** A modal window titled "Invitation Code" asks for an invitation code. The input field contains "FRGZ80XIG0". A "Continue" button is at the bottom right.
- Are you registering as a Merchant or a Service Provider?**: A question page with two radio button options:
 - Merchant:** You sell goods and/or services directly to consumers.
 - Service Provider:** You provide services to merchants and will be assisting one or more merchants with their MasterPass integration.
 A "Continue" button is at the bottom right.
- Account Registration:** A form titled "Create Your Profile". It includes fields for:
 - Language: English
 - Country of Incorporation: Germany
 - First Name: Wirecard Example
 - Last Name: Merchant
 - Phone #: DE +49 151 94424014 (Optinal)
 - Email: implementation@wirecardopen@wirecard.com
 - Confirm Email: implementation@wirecardopen@wirecard.com
 - A checkbox for "I have read and consent to the [MasterPass Operating Rules](#)".
 Buttons for "Cancel" and "Continue" are at the bottom right.
- MasterPass Setup:** A dashboard with four main sections:
 - Setup/Manage your shipping profiles.
 - Add developers.
 - Commence technical integration.
 - Approve work along the way.
 Buttons for "Start This Step" are located next to each section.

After the merchant account has been created, select **Shipping Profiles**. Merchants can have multiple shipping profiles and can also set a preferred shipping profile option.



9.7.1.2 Step 2 – Adding Wirecard as merchant integrator

In section **Developers** start the **Add Developer** process.

A third party platform provider needs to be selected.

From PSPs dropdown list please select **Wirecard Technologies GmbH** as vendor.

As a result, **Wirecard Technologies GmbH** appears as a new developer at the portal.

In section **Approval request** a new request should appear named as merchant followed by “Configuration” postfix. If the request is not displayed, please log out and log in again to merchant’s account.

Now, Wirecard will receive notifications from MasterPass, indicating that he/she has been selected to handle the technical integration of MasterPass Services on-behalf of the merchant.

9.7.1.3 Step 3 - Merchant Testing approval

The Merchant will log on to the Masterpass Merchant Portal (<https://masterpass.com/SP/Merchant/Home>), review the branding and provide approval. After clicking **Approval Requests** on the navigation bar, the user will see a list of open requests. Click **View Details** to enable **Approve** button.

After click on **Approve** button, the merchant is prompted to confirm approval.

After approval has been granted by the merchant, the developer will receive an email notification that changes have been approved. The checkout project enters the “Approved For Testing” state.

9.7.1.4 Step 4 – Merchant’s Production approval

The merchant will log on to the Masterpass Merchant Portal (<https://masterpass.com/SP/Merchant/Home>), click on **Approval Request** and click **View Details** to enable **Approve** button.

After click on **Approve** button, the merchant is prompted to confirm approval.

After merchant approval, the developer will receive email containing the merchant's Production Consumer Key and the Checkout Identifier. The checkout project has entered the final "Approved for Production" state.

Developer can log in to the PSP's account and check the checkout project.

Checkout Identifier (NOT Checkout Project ID) and Production Consumer key have to be set for merchant on Wirecard's Production environment. Please click on **View branding** to display checkout Identifier:

9.7.2 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.

9.7.3 Sample XML Request and Response for First initialization

Sample of Successful Request and Response for Pairing and Checkout

Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-
account-id>
    <request-id>ce12b82c-6399-d469-1684-ff289961b3e0</request-id>
    <transaction-type>request-checkout</transaction-type>
    <requested-amount currency="USD">4.00</requested-amount>
    <custom-fields>
        <custom-field field-name="elastic-api.merchant-origin" field-
value="http://localhost:8082/shop/masterpass/pairingredirect/">
    </custom-fields>
    <payment-methods>
        <payment-method name="masterpass"/>
    </payment-methods>
    <redirect-
url>http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/?<
/redirect-url>
    <request-type>pairing-and-checkout</request-type>
    <customer-id>unique-cust-id-005</customer-id>
</payment>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-
account-id>
    <transaction-id>81eef822-e31a-11e4-931c-00059a3c7a00</transaction-id>
```

```

<request-id>ce12b82c-6399-d469-1684-ff289961b3e0</request-id>
<transaction-type>request-checkout</transaction-type>
<transaction-state>success</transaction-state>
<completion-time-stamp>2015-04-15T10:55:17.000+08:00</completion-time-
stamp>
<statuses>
  <status code="201.0000" description="The resource was successfully
created." severity="information"/>
</statuses>
<requested-amount currency="USD">4.00</requested-amount>
<card>
  <expiration-month xsi:nil="true"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
  <expiration-year xsi:nil="true"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
</card>
<card-token/>
<custom-fields>
  <custom-field field-name="elastic-api.merchant-origin" field-
value="http://localhost:8082/shop/masterpass/pairingredirect/">
</custom-fields>
<payment-methods>
  <payment-method
url="http://testwdmp.com/engine/notification/masterpass/lightBoxPaymentP
age?
  oauth_token=9d8b2250b4ef1cb08ceba6d1e3c8656d89c6db9e&acceptable_card
s=visa,master&
  checkout_identifier=a466w4xi73dtqhzd3btie1i06e0d3bcf17&version=v6&
  callback=http%3A%2Ftestwdmp.com%2Fengine%2Fnotification%2Fmasterp
ass%3F
  payment.transaction-id%3D81eef822-e31a-11e4-931c-
00059a3c7a00%26payment.transaction-type%3Drequest-checkout%26
  payment.request-type%3Dpairing-and-checkout%26payment.customer-
id%3Dunique-cust-id-005%26
  payment.redirect-
url%3Dhttp%3A%2F%2Flocalhost%3A8082%2Fshop%2Frest%2Fmasterpass%2F
  pairingcheckoutresponse%2F%3F&lightboxurl=https%3A%2F%2Fsandbox.mast
erpass.com%2F
  lightbox%2FSwitch%2Fintegration%2FMasterPass.client.js&suppress_
ping_address=true&
  merchantOrigin=http%3A%2F%2Flocalhost%3A8082%2Fshop%2Fmasterpass%2Fp
airingredirect%2F" name="masterpass"/>
</payment-methods>
<redirect-
url>https://sandbox.masterpass.com/lightbox/Switch/integration/MasterPas
s.client.js</redirect-url>
<processing-redirect-
url>http://testwdmp.com/engine/notification/masterpass?
  payment.transaction-id=81eef822-e31a-11e4-931c-
00059a3c7a00&payment.transaction-type=request-checkout&
  payment.request-type=pairing-and-checkout&payment.customer-
id=unique-cust-id-005&
  payment.redirect-
url=<a href="http://localhost:8082/shop/rest/masterpass/pairingcheckoutrespon
se/?&
  lightboxurl=https://sandbox.masterpass.com/lightbox/Switch/integrati
on/MasterPass.client.js&
  suppress_shipping_address=true&merchantOrigin=http://localhost:8082/
shop/masterpass/pairingredirect/
</processing-redirect-url>
<wallet>
  <merchant-id>a466w4xi73dtqhzd3btie1i06e0d3bcf17</merchant-id>

```

```

<request-token>9d8b2250b4ef1cb08ceba6d1e3c8656d89c6db9e</request-
token>
  <pair-token>73239d70603890eac173fe59d3e0422da0f475b3</pair-token>
</wallet>
<request-type>pairing-and-checkout</request-type>
<customer-id>unique-cust-id-005</customer-id>
</payment>

```

Sample of Successful Notification

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-
account-id>
  <transaction-id>81eef822-e31a-11e4-931c-00059a3c7a00</transaction-id>
  <request-id>ce12b82c-6399-d469-1684-ff289961b3e0-request-
checkout</request-id>
  <transaction-type>request-checkout</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-04-15T11:04:45.000+08:00</completion-time-
stamp>
  <statuses>
    <status code="201.0000" description="masterpass:The resource was
successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">4</requested-amount>
  <account-holder>
    <first-name>Kim</first-name>
    <last-name>Li</last-name>
    <email>kimmy@email.com</email>
    <phone>1-5234125162</phone>
    <address>
      <street1>tampines</street1>
      <street2/>
      <city>singapore</city>
      <country>SG</country>
      <postal-code>971649</postal-code>
    </address>
  </account-holder>
  <card>
    <expiration-month>2</expiration-month>
    <expiration-year>2018</expiration-year>
  </card>
  <card-token>
    <token-id>5709141208470014</token-id>
    <masked-account-number>520474*****0014</masked-account-number>
  </card-token>
  <custom-fields>
    <custom-field field-name="provider.response.wallet.id" field-
value="101"/>
    <custom-field field-name="elastic-api.merchant-origin"
      field-
value="http://localhost:8082/shop/masterpass/pairingredirect/">
    </custom-fields>
    <payment-methods>
      <payment-method
        name="masterpass"
        url="http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/
?">
    </payment-methods>
    <api-id>---</api-id>
  </custom-fields>
</payment>

```

```

<processing-redirect-
url>http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/?
</processing-redirect-url>
  <cancel-redirect-
url>http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/?
</cancel-redirect-url>
  <fail-redirect-
url>http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/?
</fail-redirect-url>
  <success-redirect-
url>http://localhost:8082/shop/rest/masterpass/pairingcheckoutresponse/?
</success-redirect-url>
  <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>Gz21t4rUrCSK6Llhk3qANZGHEcQ=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>fDkDrp1b65oHZk0H6R2RW3iOffP+9LH9kLoJ0qUucDjKE/S+MxwSuRwmQ/7n2tS0+y8e6K+gv9UV
          NEa9JH2R0wxx8903GbyjbkEHBCNXRJqWc2NtrxgCbjhzq3s3Y2iU3Q2MGkPqOT2zwPzRys3AEJ1
          Tzf7U/IWt1O2DFiWDx1Z8SPOfN5oKgsFYJ9U+SAThSi57qj3wjJVzaGl4lCd2km7M5DZDgWbNfsf
          zoNoiJHpK208exxdxZh/gSljCKkqkgst/UqA3TsmHtgwfBXdidrofL2s1v1K/cvL4tev7z4en0wp
          HNDB2Qn20y822J958qPhRuZf4t2aNdCGNHME6A==</SignatureValue>
        <KeyInfo>
          <X509Data>
            <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName>
            <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UE
              CBMCT04xE DAOBgNVBAcTB1Rvc9udG8xIjAgBgNVBAoTGVdpVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hb9qIFNhaHUwHhcNMTEwNjI0MDQ0
              NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xE DAOBgNVBAcT
              B1Rvc9udG8xIjAgBgNVBAoTGVdpVjYXJkIEVsYXN0aWMgUGF5bWVudHMxEzARBgNVBAAsTCk9w
              ZXJhdGlvbnMxEzARBgNVBAMTCk1hb9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQ
              UAA4IBDwAwggEK
              AoIBAQCc8rTt4N5fNeVzlSrgOXKDE2YUSfJx7xBozFZ3Vh3XQyy3IpIuEfZz7004k4HeonfTxCN
              etBvJ9rgNc0Cxrk/euMj3pOUR9WYN2eAXC0r5pUIAZhIAAnSxUSAif3JKBxf7gDAik5d8RT5HaJV
              4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3IyLwhDbYIyU6j4fMKyH1AMGzW7VgKD
              2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvNRWbIzyQuyUyDeUb
            </X509Data>
          </KeyInfo>
        </Signature>
      </processing-redirect>
    
```

```

        QmhVQKL0W1Mb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAEwDQYJKoZIhv
cNAQEFBQADggEB
        ADgkuN/e2IFY7JXdbjNJbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2
pbOD3HBXoyovZr
        mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SS
ucBfIS5TJ54e7m
        02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8K
CHpfw2j8G+dDgU
        jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFF
oJNdG13AzYwDP3
        x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
</payment>

```

Pairing and Checkout Lightbox initiation

Initialize MasterPass lightbox with the following script in the origin_url page

```

MasterPass.client.checkout({
    "requestToken": "${payment.wallet.requestToken}",
    "callbackUrl": "${payment.processingRedirectUrl}",
    "merchantCheckoutId": "${payment.wallet.merchantId}",
    "allowedCardTypes": ["visa,master"],
    "suppressShippingAddressEnable": "true",
    "pairingRequestToken": "${payment.wallet.pairToken}",
    "requestPairing": true,
    "requestedDataTypes": ["ADDRESS,PROFILE,CARD"],
    "version": "v6"
});

```

Sample of Successful Request and Response for Pairing (First initialization)

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-
account-id>
    <request-id>af68f354-b659-f32f-8a5c-de6f13c2c68e</request-id>
    <transaction-type>request-checkout</transaction-type>
    <requested-amount currency="USD">4.00</requested-amount>
    <custom-fields>
        <custom-field field-name="elastic-api.merchant-origin" field-
value="http://testwdmp.com/shop/masterpass/pairingredirect/">
        </custom-fields>
        <payment-methods>
            <payment-method name="masterpass"/>
        </payment-methods>
        <redirect-
url>http://testwdmp.com/shop/rest/masterpass/requestcheckoutresponse?</r
edirect-url>
        <request-type>pairing</request-type>
        <customer-id>unique-cust-id-003</customer-id>
    </payment>

```

Response

```

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

```

```

<merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-
account-id>
  <transaction-id>c4f6d43f-d6bb-11e4-aec6-3c970e23ad17</transaction-id>
  <request-id>af68f354-b659-f32f-8a5c-de6f13c2c68e</request-id>
  <transaction-type>request-checkout</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-03-30T17:05:32.000+08:00</completion-time-
stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully
created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">4.00</requested-amount>
  <card>
    <expiration-month xsi:nil="true">
      <ns1:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns1:Nil>
      </ns1:xsi>
    </expiration-month>
    <card-token xsi:nil="true">
      <ns1:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns1:Nil>
      </ns1:xsi>
    </card-token>
    <custom-fields>
      <custom-field field-name="elastic-api.merchant-origin" field-
value="http://testwdmp.com/shop/masterpass/pairingredirect"/>
    </custom-fields>
    <payment-methods>
      <payment-method
        url="http://testwdmp.com/engine/notification/masterpass/
          lightBoxPaymentPage?oauth_token=1bfb2a3ab5ffb80970647e6b5eb856626f7
          600b&acceptable_cards=visa,master&
          checkout_identifier=a466w4xi73dtqhzd3btie1i06e0d3bcf17&version=v6&
          callback=http%3A%2F%2Ftestwdmp.com%2Fengine%2Fnotification%2Fmasterp
          ass%3F
          payment.transaction-id=c4f6d43f-d6bb-11e4-aec6-
          3c970e23ad17%26payment.transaction-type=request-checkout%26
          payment.request-type=pairing%26payment.customer-id=unique-cust-id-
          003%26
          payment.redirect-
        url=http%3A%2F%2Flocalhost%3A8082%2Fshop%2Frest%2Fmasterpass%2Frequestch
        eckoutresponse%3F&
          lightboxurl=https%3A%2F%2Fsandbox.masterpass.com%2Flightbox%2FSwitch
          %2Fintegration%2FMasterPass.client.js&
          suppress_shipping_address=true&merchantOrigin=http%3A%2F%2Flocalhost
          %3A8082%2Fshop%2Fmasterpass%2Fpairingredirect%2F" name="masterpass"/>
      </payment-methods>
      <redirect-
        url="http://testwdmp.com/shop/rest/masterpass/requestcheckoutresponse?
        redirect-url>
      <wallet>
        <merchant-id>a466w4xi73dtqhzd3btie1i06e0d3bcf17</merchant-id>
        <pair-token>1bfb2a3ab5ffb80970647e6b5eb856626f7600b</pair-token>
      </wallet>
      <request-type>pai
      <customer-id>unique-cust-id-003</customer-id>
    </payment>
  
```

Pairing Lightbox initiation

requestExpressCheckout is required only if merchant is enabled for Express Checkout.

```
MasterPass.client.connect({
```

```

    "pairingRequestToken": "${payment.wallet.pairToken}",
    "callbackUrl": "${payment.processingRedirectUrl}",
    "merchantCheckoutId": "${payment.wallet.merchantId}",
    "requestedDataTypes": ["ADDRESS, PROFILE, CARD"],
    "requestPairing": true,
    "requestExpressCheckout": true
} );

```

9.7.4 Sample XML Request and Response for Subseqent initialization

Sample of Successful Request and Response for Precheckout

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-account-
  id>
  <request-id>a930e573-37f5-4866-0f39-34693f4e95ef</request-id>
  <transaction-type>precheckout</transaction-type>
  <requested-amount currency="USD">4.00</requested-amount>
  <custom-fields>
    <custom-field field-name="elastic-api.merchant-origin" field-
    value="http://testwdmp.com/shop/masterpass/precheckout/">
  </custom-fields>
  <payment-methods>
    <payment-method name="masterpass"/>
  </payment-methods>
  <customer-id>unique-cust-id-005</customer-id>
</payment>

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-account-
  id>
  <transaction-id>2d6be9c8-d6ae-11e4-aec6-3c970e23ad17</transaction-id>
  <request-id>a930e573-37f5-4866-0f39-34693f4e95ef</request-id>
  <transaction-type>precheckout</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-03-30T15:28:12.000+08:00</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">4.00</requested-amount>
  <card>
    <expiration-month xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    <expiration-year xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
  </card>
  <card-token/>
  <custom-fields>
    <custom-field field-name="elastic-api.merchant-origin" field-
    value="http://testwdmp.com/shop/masterpass/precheckout/">
  </custom-fields>
  <payment-methods>
    <payment-method name="masterpass"/>
  </payment-methods>
  <wallet>
    <merchant-id>a466w4xi73dtqhzd3btie1i06e0d3bcf17</merchant-id>

```

```

<provider-ref>a4a6x55-cxth67-i7s0m6ir-1-i7vk0ck6-1lye</provider-ref>
<wallet-
data>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRLTgiIHN0YW5kYWxbmU9
InlcyI/PjxQcmVjaGVja291dERhdGE+PENhcmRzPjxDYXJkPjxCcmFuZE1kPm1hc3R1cjwvQ
nJhbmRJZD48QnJhbmROYW11Pk1hc3R1ckNhcmQ8L0JyYW5kTmFtZT48Q2FyZEhbGR1ck5hbW
U+S2ltIEpcPC9DYXJkSG9sZGVyTmFtZT48RXhwaXJ5TW9udGg+MjwvRXhwaXJ5TW9udGg+PEV
4cG1yeV1lYXI+MjAxODwvRXhwaXJ5WWVhcj48Q2FyZE1kPjE2NjU5OTgxNzvQ2FyZE1kPjxM
YXN0Rm91cj4wMDE0PC9MYXN0Rm91cj48Q2FyZEfsaWFzPnR1c3QtMTwvQ2FyZEfsaWFzPjxTZ
Wx1Y3R1ZEFzRGVmYXVsD50cnV1PC9TZWx1Y3R1ZEFzRGVmYXVsD48L0NhcmQ+PC9DYXJkcZ
48Q29udGFjdD48Rmlyc3ROYW11PktcbTwvRmlyc3ROYW11PjxMYXN0TmFtZT5MaTwvTGFzdE5
hbWU+PENvdW50cnk+VVM8L0NvdW50cnk+PEVtYW1sQWRkcmVzcz5raW1teUB1bWFpbC5jb208
L0VtYW1sQWRkcmVzcz48UGHvbmVOdW1iZXI+MS01MjM0MTI1MTYyPC9QaG9uZU51bWJ1cj48L
0NvbnRhY3Q+PFN0aXBwaW5nQWRkcmVz2VzPjxTaGlwcGluZ0FkZHJ1c3M+PENpdHk+c2luZ2
Fwb3J1PC9DaXR5PjxDb3VudHJ5P1NHPC9Db3VudHJ5PjxDb3VudHJ5U3ViZG12aXNpb24+c2c
8L0NvdW50cn1TdWJkaXZpc21vbj48TGluzTE+dGftcGluzXM8L0xpbmUxPjxMaW51Mj48L0xp
bmUyPjxMaW51Mz48L0xpbmUzPjxQb3N0YWxDb2R1Pjk3MTY0OTwvUG9zdGfsQ29kZT48UmVja
XBpZW50TmFtZT5LaW0gTGk8L1J1Y21waWVudE5hbWU+PFJ1Y21waWVudFBob251TnVtYmVyPj
EtNTIzNDEyNTE2MjwvUmVjaXBpZW50UGHvbmVOdW1iZXI+PEFkZHJ1c3NJZD4xNjY1OTk4MzI
8L0FkZHJ1c3NJZD48U2VsZWN0ZWRBc0R1ZmF1bHQ+dHJ1ZTwvU2VsZWN0ZWRBc0R1ZmF1bHQ+
PC9TaGlwcGluZ0FkZHJ1c3M+PC9TaGlwcGluZ0FkZHJ1c3N1cz48V2FsbGV0TmFtZT5Nb2Jpb
GU8L1dhbGx1dE5hbWU+PFByZWN0ZWNrb3V0VHJhbnNhY3Rpb25JZD5hNGE2eDU1LWN4dGg2Ny
1pN3MwbTZpci0xLWk3dmswY2s2LTFseWU8L1ByZWN0ZWNrb3V0VHJhbnNhY3Rpb25JZD48Q29

uc3VtZXJXYWxsZXRJZD4xNjY1OTcyNDI8L0NvbnN1bWVvY2FsbGV0SWQ+PC9QcmVjaGVja291dERhdG
E+
    </wallet-data>
</wallet>
<customer-id>unique-cust-id-005</customer-id>
</payment>
```

Sample of Successful Request and Response for Connect Checkout

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-account-
id>
    <request-id>9d643ec9-1bab-b6c0-0e1c-ea4f4bd38720</request-id>
    <transaction-type>precheckout</transaction-type>
    <requested-amount currency="USD">4.00</requested-amount>
    <custom-fields>
        <custom-field field-name="elastic-api.merchant-origin">
            <field-value>http://localhost:8082/shop/masterpass/precheckout/</field-value>
        </custom-field>
    </custom-fields>
    <payment-methods>
        <payment-method name="masterpass"/>
    </payment-methods>
    <customer-id>unique-cust-id-005</customer-id>
</payment>
```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-account-
id>
    <transaction-id>4c4a9e5d-d83b-11e4-aec6-3c970e23ad17</transaction-id>
    <request-id>9d643ec9-1bab-b6c0-0e1c-ea4f4bd38720</request-id>
    <transaction-type>precheckout</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-04-01T14:51:27.000+08:00</completion-time-stamp>
```

```

<statuses>
  <status code="201.0000" description="The resource was successfully created." severity="information"/>
</statuses>
<requested-amount currency="USD">4.00</requested-amount>
<card>
  <expiration-month>xi</expiration-month>
  <expiration-year>2001</expiration-year>
</card>
<card-token/>
<custom-fields>
  <custom-field field-name="elastic-api.merchant-origin" field-value="http://localhost:8082/shop/masterpass/precheckout/">
</custom-fields>
<payment-methods>
  <payment-method name="masterpass"/>
</payment-methods>
<wallet>
  <merchant-id>a466w4xi73dtqhzd3btie1i06e0d3bcf17</merchant-id>
  <provider-ref>a4a6x55-2upnfm-i7xeb5e-1-i7ydktwm-v6u</provider-ref>
  <wallet-data>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRLTgiIHN0YW5kYWxvbmU9InlcyIPjxQcmVjaGVja291dERhdGE+PENhcmRzPjxDYXJkPjxCcmFuZE1kPm1hc3RlcjwvQnJhbhRJZD48QnJhbhROYW11Pk1hc3RlcNhcmQ8L0JyYW5kTmFtZT48Q2FyZhbGRlc5hbWU+S21tIExpPC9DYXJkSG9sZGVyTmFtZT48RKhwaXJ5TW9udGg+MjwvRxhwaXJ5TW9udGg+PEV4cGlyeV11YXI+MjAxODwvRxhwaXJ5WWVhcj48Q2FyZE1kPjE2NjU5OTgxNzwvQ2FyZE1kPjxMYXN0Rm91cj4wMDE0PC9MYXN0Rm91cj48Q2FyZEfsaWFzPnR1c3QtMTwvQ2FyZEfsaWFzPjxTZWx1Y3R1ZEFzRGVmYXVsD50cnv1PC9TZWx1Y3R1ZEFzRGVmYXVsD48L0NhcmQ+PC9DYXJkc48Q29udGFjdD48Rmlyc3ROYW11PktbTwvRmlyc3ROYW11PjxMYXN0TmFtZT5MaTwvTGFzdE5hbWU+PENvdW50cnk+VVM8L0NvdW50cnk+PEVtYwlsQWRkcmVzc5raW1teUB1bWFpbC5jb208L0VtYwlsQWRkcmVzc48UGHvbmV0dW1iZXI+MS01MjM0MTI1MTYyPC9QaG9uZU51bWJlcj48L0NvbhRy3Q+PFNoaXBwaW5nQWRkcmVzc2VzPjxTaGlwcGluZ0FkZHJ1c3M+PENpdHk+c21uZ2Fwb3J1PC9DaXR5PjxDb3VudHJ5P1NHPC9Db3VudHJ5PjxDb3VudHJ5U3VizG12aXNpb24+c2c8L0NvdW50cn1TdWJkaXZpc21vbj48TGluzTE+dGftcGluZXM8L0xpbmUxPjxMaW51Mj48L0xpmbUyPjxMaW51Mz48L0xpmbUzPjxQb3N0YWxDb2R1Pjk3MTY0OTwvUG9zdGFsQ29kZT48UmVjaXBpZW50TmFtZT5LaW0gTGk8L1J1Y21waWVudE5hbWU+PFJ1Y21waWVudFBob251TnVtYmVypjEtNTizNDEyNTE2MjwvUmVjaXBpZW50UGHvbmV0dW1iZXI+PEFkZHJ1c3NJZD4xNjY1OTk4MzI8L0FkZHJ1c3NjZD48U2VsZWN0ZWRBc0R1ZmF1bHQ+dHJ1ZTwvU2VsZWN0ZWRBc0R1ZmF1bHQ+PC9TaGlwcGluZ0FkZHJ1c3M+PC9TaGlwcGluZ0FkZHJ1c3N1cz48V2FsbGV0TmFtZT5Nb2JpbGU8L1dhbGxldE5hbWU+PFByzWN0zWNrb3V0VHJhbnNhY3Rpb25JZD5hNGE2eDU1LTJ1cG5mbS1pN3h1Z2I1ZS0xLWk3eWRrdHdtLXY2dTtwvUHJ1Y2h1Y2tvdxRUCmfuc2Fjdg1vbk1kPjxDb25zdW11cldhbGxldElkPjE2NjU5NzI0MjwvQ29uc3VtZXJXYWxsZXRJZD48L1ByZWN0ZWNrb3V0RGF0YT4=</wallet-data>
</wallet>
<customer-id>unique-cust-id-005</customer-id>
</payment>

```

Sample of Successful Request and Response for Express Checkout

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-account-id>
  <request-id>a66d59f8-a659-e8fe-bee7-715bcb9f1781</request-id>
  <transaction-type>request-checkout</transaction-type>
  <requested-amount currency="USD">4.00</requested-amount>

```

```

<parent-transaction-id>b30f109c-d6b7-11e4-aec6-3c970e23ad17</parent-
transaction-id>
<payment-methods>
  <payment-method name="masterpass"/>
</payment-methods>
<redirect-
url>http://testwdmp.com/shop/rest/masterpass/requestcheckoutresponse?</redire
ct-url>
<wallet>
  <card-id>154131309</card-id>
  <address-id>154132238</address-id>
  <provider-ref>a4a6x55-cxth67-i7s0m6ir-1-i7vmg0wz-1pg8</provider-ref>
</wallet>
<request-type>express-checkout</request-type>
<customer-id>unique-cust-id-003</customer-id>
</payment>

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>45119325-d77c-4c80-bd5b-8f44bda734b9</merchant-
account-id>
  <transaction-id>e64a07b8-d6b7-11e4-aec6-3c970e23ad17</transaction-id>
  <request-id>a66d59f8-a659-e8fe-bee7-715bcb9f1781</request-id>
  <transaction-type>request-checkout</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-03-30T16:37:51.000+08:00</completion-time-
stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">4.00</requested-amount>
  <parent-transaction-id>b30f109c-d6b7-11e4-aec6-3c970e23ad17</parent-
transaction-id>
  <card>
    <expiration-month>1</expiration-month>
    <expiration-year>2022</expiration-year>
  </card>
  <card-token>
    <token-id>5709141208470014</token-id>
    <masked-account-number>520474*****0014</masked-account-number>
  </card-token>
  <custom-fields>
    <custom-field field-name="elastic-api.merchant-origin" field-
value="http://testwdmp.com/shop/masterpass/precheckout/"></custom-field>
  </custom-fields>
  <payment-methods>
    <payment-method name="masterpass"/>
  </payment-methods>
  <api-id>---</api-id>
  <redirect-
url>http://testwdmp.com/shop/rest/masterpass/requestcheckoutresponse?</redire
ct-url>
  <wallet>
    <card-id>154131309</card-id>
    <address-id>154132238</address-id>
    <provider-ref>a4a6x55-cxth67-i7s0m6ir-1-i7vmg0wz-1pg8</provider-ref>
  </wallet>

```

```
<request-type>express-checkout</request-type>
<customer-id>unique-cust-id-003</customer-id>
</payment>
```

Sample of Successful Request and Response for Debit (**After Pairing and Checkout / Connect Checkout / Express Checkout**)

Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-
account-id>
  <request-id>71867a2d-ec91-6dea-77a1-8b6b4ae365e5</request-id>
  <transaction-type>debit</transaction-type>
  <requested-amount currency="USD">4.00</requested-amount>
  <parent-transaction-id>9d3fd50b-e274-11e4-931c-00059a3c7a00</parent-
transaction-id>
  <card-token>
    <token-id>5709141208470014</token-id>
  </card-token>
  <payment-methods>
    <payment-method name="masterpass"/>
  </payment-methods>
  <wallet>
    <provider-ref>a4a6x55-fkd5a2-i8acyfmv-1-i8gyvqb1-8jhd</provider-ref>
    <provider-transaction-id>a4a6x55-fkd5a2-i8acyfmv-1-i8gyvqb1-
8jhd</provider-transaction-id>
  </wallet>
</payment>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>1c13c26e-f754-4fb9-9082-f6025cd26bc5</merchant-
account-id>
  <transaction-id>1a22157a-e275-11e4-931c-00059a3c7a00</transaction-id>
  <request-id>71867a2d-ec91-6dea-77a1-8b6b4ae365e5</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-04-14T15:11:03.000+08:00</completion-time-
stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">4.00</requested-amount>
  <parent-transaction-id>9d3fd50b-e274-11e4-931c-00059a3c7a00</parent-
transaction-id>
  <card>
    <expiration-month>xi:nil="true">
    <expiration-year>xi:nil="true">
  </card>
  <card-token>
    <token-id>5709141208470014</token-id>
  </card-token>
```

```

<custom-fields>
  <custom-field      field-name="elastic-api.merchant-origin"      field-
value="http://localhost:8082/shop/masterpass/precheckout/">/>
</custom-fields>
<payment-methods>
  <payment-method name="masterpass"/>
</payment-methods>
<api-id>---</api-id>
<wallet>
  <provider-ref>a4a6x55-fkd5a2-i8acyfmv-1-i8gyvqb1-8jhd</provider-ref>
  <provider-transaction-id>a4a6x55-fkd5a2-i8acyfmv-1-i8gyvqb1-
8jhd</provider-transaction-id>
</wallet>
<provider-transaction-reference-id>75877512066328602591</provider-
transaction-reference-id>
</payment>

```

9.8 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 6.4).

9.8.1 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	<p>This is the amount of the transaction.</p> <p>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.</p>
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 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 redirected 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 redirected 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 redirected 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.

9.8.2 Debit for Carrier Billing

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

9.8.3 XML- Samples for Carrier Billing

Sample of a successful debit request.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>d97a261d-dbee-4993-b323-2349d51b768b</merchant-account-id>
<requested-amount currency="EUR">10</requested-amount>
<request-id>65b308cd-f2ef-4b42-8f6a-434c7e117d5f</request-id>
<transaction-type>debit</transaction-type>
<payment-methods>
  <payment-method name="carrier-billing"/>
</payment-methods>
<account-holder>
  <address>
    <country>SK</country>
  </address>
</account-holder>
<consumer-id>consumer_i2345</consumer-id>
</payment>
```

Sample of a successful debit response.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>d97a261d-dbee-4993-b323-2349d51b768b</merchant-account-id>
<transaction-id>68cae486-2633-11e5-94a1-0050b65c678c</transaction-id>
<request-id>65b308cd-f2ef-4b42-8f6a-434c7e117d5f</request-id>
<transaction-type>debit</transaction-type>
<transaction-state>success</transaction-state>
<completion-time-stamp>2015-07-09T14:09:55.000+02:00</completion-time-stamp>
<statuses>
  <status code="201.0000" description="The resource was successfully created." severity="information"/>
</statuses>
<requested-amount currency="EUR">10</requested-amount>
<account-holder>
  <address>
    <country>SK</country>
  </address>
</account-holder>
<payment-methods>
  <payment-method url="https://buy.boku.com/checkoutidentify/72b8e0eb061f74255038fdd0/buy.js"
name="carrier-billing"/>
</payment-methods>
<consumer-id>consumer_i2345</consumer-id>
</payment>
```

After the successful response, buyer is redirected to Boku to proceed the payment (in this case it is Boku sandbox). Following is a successful payment notification received by merchant.

```
response_signature=2a715f3ac100ad38906d48c84717840c40f6a0990390c8be0273cb23104d7960&phone=SK00&t
ransaction_type=debit&locale=&completion_time_stamp=20150709141336&status_code_1=201.0000&status_sev
erity_1=information&transaction_state=success&transaction_id=ec87fe6b-2633-11e5-94a1-
0050b65c678c&country=SK&merchant_account_id=d97a261d-dbee-4993-b323-
2349d51b768b&ip_address=127.0.0.1&provider_transaction_reference_id=&request_id=5ebb92fc-b72d-478c-
98ec-
7aca869b1e4c&requested_amount=15.00&requested_amount_currency=EUR&status_description_1=boku%3AThe
+resource+was+successfully+created.&provider_transaction_id_1=&authorization_code=&
```

9.8.4 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.

9.8.5 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 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.

9.8.6 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-ae62cdffadaf1</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-ae62cdffadaf1</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-ae62cdffad1</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+sdezj7ZUttkLOQxxcD4NmnnUcVtFtDnZc73vjWSTL6pBsfl9vXMGhqqoeAzaGpTlbEKQ5rm4Z7oKfWnjPiO4rQ4Cj4uUli8EqCBvqCOE6NRFOkFNvWBxYOLx0wEphVxf+xtQ3oJiw+8tbmbg7U9fUVKTWlmyJ2jMOf+1gaKxWCBS7pUTX5qAK/o7rRglo5DBFOhql2eqyOkB8V0rsYUBLflSJdeOlh4iQ1DTNhXXr5VVNEd1Fknda+APut6gpQ3rWM+UOYypXMIMaLj2UPMTv3PjeL7C+dNVJ+15qRfc0QL+9l/oVL5iZfO3tFKw==</SignatureValue>
<KeyInfo>
  <X509Data>
    <X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Payments,L=Toronto,ST=ON,C=CA</X509SubjectName> Elastic
    <X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UE
    CBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpVjYXJkIEVsYXN0aWMgUGF5bVVu
    dHMxEzARBgNVBAAsTCk9wZXJhdGlvbMxEzARBgNVBAMTck1hbm9qIFNhaHUwHhcNMTEwNj0MDQ0
    NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcT
    B1Rvcn9udG8xIjAgBgNVBAoTGVdpVjYXJkIEVsYXN0aWMgUGF5bVVudHMxEzARBgNVBAAsTCk9w
    ZXJhdGlvbMxEzARBgNVBAMTck1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
    AoIBAQCc8rTt4N5fNeVzlsRgOKXDE2YUSfjx7xBozFZ3Vh3XQyy3IpluEfZz7004k4HeonfTxCN
    etBvJ9rgNc0Cxrk/euMj3pOURE9WYN2eAXC0r5pUIAZhIAAnSxUSAf3JKBxf7gDAik5d8RT5HaJV
    4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3lyLwhDbYlyU6j4fMKyHIIAMGzW7VgKD
    2pqU6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvRWblzyQuyUyDeUb
    QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB
    ADgkuN/e2IFy7JXdbjNjbKbd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr
    mk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/Whsx89SSucBfIS5TJ54e7m
    02qvGoK8UA/IRblQ6DZ9hEKV5VQKiMx3ubwwHGxfOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU
    jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRKO/sJtZBzQajpnsZ9NFf0JNdG13AzYwDP3
    x/QspK0jYn1KZw1qz524VWoQuer8Xj30A2jntA=</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>
```

Sample of a failed Carrier Billing transaction response:

```

<?xml version="1.0" encoding="UTF-8"?>
<payment      xmlns="http://www.elastic-payments.com/schema/payment"><merchant-account-
id>d97a261d-dbee-4993-b323-2349d51b768b</merchant-account-id><transaction-id>5c0fed8d-14ef-
11e5-a5ba-0050b667f65b</transaction-id><request-id>96d8c249-9fb0-4fc8-9f50-
97f72d41d2d2</request-id><transaction-type>debit</transaction-type><transaction-
state>failed</transaction-state><completion-time-stamp>2015-06-17T12:49:58.000Z</completion-time-
stamp><statuses><status  severity="error"  description="boku:insufficient  funds/  no  funds"
code="500.2401"/></statuses><requested-amount currency="EUR">1.00</requested-amount><account-
holder><first-name>John</first-name><last-name>Doe</last-
name><email>john.doe@test.com</email><phone>SK00</phone>
<address><street1>123
anystreet</street1><city>Brantford</city><country>SK</country></address>
</account-holder><order-detail>nekupuj glupociny</order-detail><payment-methods><payment-method
name="carrier-billing"/></payment-methods><consumer-id>consumer_i2345</consumer-id><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>7Lo+9CbMzuQ4M6Sdyg7NCwZ7Ur4=</D
igestValue></Reference></SignedInfo><SignatureValue>XzmJf/sdNPihQlxsd6skhMScmN8xYN10N96XaYsxzwOAUNjP
NfM7M1Wui2tEN64h1ZeBgMLu/CmlA
uA+43pnqF6fQwQz3SzwsGBujM5wD+RRwp9EoEqN1zbIoT+BXSVu3b1sOFUIP7j0OQsbuk71aXdc
```

```
dLbTmdEK/nDd+CF/jEm0h8pOcqjmwPMH/t/L9o/QzK556aZuTmlmUkJErSOXbXvquAUUs3hQB0Sb4
IIGR9BWi8BH4lpnodZz2nDo/ecKwo1U/bFPQeMYowifrqDndCOtCf38S/q1FKbRRzpk8t9KwNDlz
Lm5+raC55dTTwoH60mFj9eMCeNsKHfkbbVfWriA==</SignatureValue><KeyInfo><X509Data><X509SubjectName>CN
=Manoj Sahu,OU=Operations,O=Wirecard Elastic
Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqh
kiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UE
CBMCT04xEDAOBgNVBAcTB1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bVVu
dHMxEzARBgNVBAstCk9wZXJhdGlvbNmxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0
NDA5WhcNMTQwMzlwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcT
B1Rvcm9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bVVudHMxEzARBgNVBAstCk9w
ZXJhdGlvbNmxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xXBozFZ3Vh3XQyy3IpluEfZz7004k4HeonfTxCN
etBvJ9rgNc0Cxrk/euMj3pOUR9WYN2eAXC0r5pUIAZhIAAnSxUsa1F3JKBxf7gDAik5d8RT5HaJV
4n5cXJQ/uhAEYU3EGN/74UrD2Us0YD3VBXTJS5VgSi/c3lyLwhDbYlyU6j4fMKyHIIAMGzW7VgKD
2pqub6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvnrWbIzyQuyUyDeUb
QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZlhcNAQEFBQADggEB
ADgkuN/e2IFy7JXdbjNjbKbd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr
mk/KqOyUL+unVcR+APjxX4KP25sdklglmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBflS5TJ54e7m
02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU
jlp9LvjVNmJzfNBbk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtZBzQajpnsZ9NFoJNdG13AzYwDP3
x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate></X509Data></KeyInfo></Signature></payment>
```

Sample of a failed Carrier Billing notification:

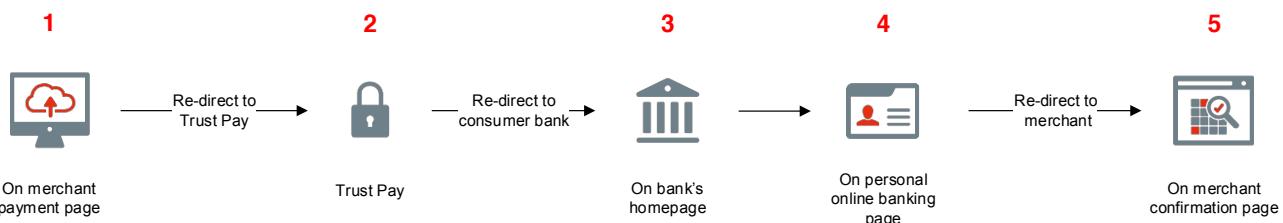
```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
-<payment self="http://localhost:8080/engine/rest/merchants/d97a261d-dbee-4993-
b323-2349d51b768b/payments/5c0fed8d-14ef-11e5-a5ba-0050b667f65b"
xmlns="http://www.elastic-payments.com/schema/payment"><merchant-account-id
ref="http://localhost:8080/engine/rest/merchants/d97a261d-dbee-4993-b323-
2349d51b768b">d97a261d-dbee-4993-b323-2349d51b768b</merchant-account-
id><transaction-id>5c0fed8d-14ef-11e5-a5ba-0050b667f65b</transaction-id><request-
id>96d8c249-9fb0-4fc8-9f50-97f72d41d2d2</request-id><transaction-
type>debit</transaction-type><transaction-state>failed</transaction-
state><completion-time-stamp>2015-06-17T12:49:58.000+02:00</completion-time-
stamp>
<statuses><status severity="error" description="insufficient funds/ no funds"
code="500.2401"/><status severity="error" description="Duplicate notification
received from the bank. No action needed." code="400.1100"/></statuses>
<requested-amount currency="EUR">1.000000</requested-amount><parent-transaction-
id>1d93d6af-14ef-11e5-a5ba-0050b667f65b</parent-transaction-id>-
<account-holder><first-name>John</first-name><last-name>Doe</last-
name><email>john.doe@test.com</email><phone>SK00</phone>-<address><street1>123
anystreet</street1><city>Brantford</city><country>SK</country></address></account-
holder><order-detail>Hogfather - book</order-detail>-<payment-methods><payment-
method name="carrier-billing"/></payment-methods><consumer-
id>consumer_i2345</consumer-id></payment>
```

9.9 Trust Pay

TrustPay offers reliable, safe and fast payment processing not only for e-commerce merchants. With one single integration of the TrustPay, you receive instant wire transfer payments from your end users/customers.

There is no need for the end-user/customers to own a TrustPay account – this is an instant wire transfer through the internet banking of his/her local bank. The current flow is with two redirects:

- a) when a buyer is on merchant's page, he gets a redirect URL to Trust Pay where he selects his country of origin and a bank,
- b) then he is redirected to a payment site of his bank to proceed the payment



9.9.1 Fields for Trust Pay

Term	Man /Opt	Format	Max Size	Description
account-holder.address.country	O	Alphanumeric	2	Country identifier of a payer.
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	18,2	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 "trustpay".
locale	O	Alphanumeric	2	Country identifier of a payer.
requested-amount currency	M	Alphanumeric	3	Currency unit.

9.9.2 XML samples for a debit transaction

Below is a GET-URL transaction with redirect URL to confirm debit.

Request with specific amount and payment method

```

POST http://127.0.0.1:8080/engine/rest/paymentmethods/?payment-method-name=trustpay HTTP/1.1
Content-Type: application/xml; charset=UTF-8
Content-Length: 412
  
```

```

Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.3.1 (java 1.5)
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
  <request-id>02f3421f-de07-4491-bf53-98118de936f3</request-id>
  <transaction-type>debit</transaction-type>
    <payment-methods>
      <payment-method name="trustpay"/>
    </payment-methods>
  <requested-amount currency="EUR">10.11</requested-amount>
  <account-holder>
    <address>
      <country>SK</country>
    </address>
  </account-holder>
</payment>

```

Response with redirect URL

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
  <transaction-id>ee3870bb-fa19-11e4-a14a-0050b65c678c</transaction-id>
  <request-id>02f3421f-de07-4491-bf53-98118de936f3</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-05-14T11:17:07.000+02:00</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10.11</requested-amount>
  <payment-methods>
    <payment-method
url="https://ib.test.trustpay.eu/mapi/pay.aspx?AID=2107796749&AMT=10.11&CUR=EUR&REF=ee3
870bb-fa19-11e4-a14a-
0050b65c678c&URL=http%3A%2F%2F127.0.0.1%3A8080%2Fengine%2Fnotification%2Ftrustpay%2F%2Fredire
ct%2F&NURL=http%3A%2F%2F127.0.0.1%3A8080%2Fengine%2Fnotification%2Ftrustpay%2F&SIG=20A07
4A8DBBDD06D03D0693C8E281E03CDD10123A33202B279AEAE228106F7D" name="trustpay"/>
  </payment-methods>
</payment>

```

Notification for a success debit.

```

GET
http://127.0.0.1:8080/engine/notification/trustpay?AID=2107796749&TYP=CRDT&AMT=10.11&CUR=EUR&REF=ee
3870bb-fa19-11e4-a14a-
0050b65c678c&RES=0&TID=12345&OID=12345&TSS=Y&SIG=5B1351487695448DDDCEC350260260A8D29A5E835D
D769653985AB601B880BC0 HTTP/1.1
Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.3.1 (java 1.5)

```

A sample request for failed scenario.

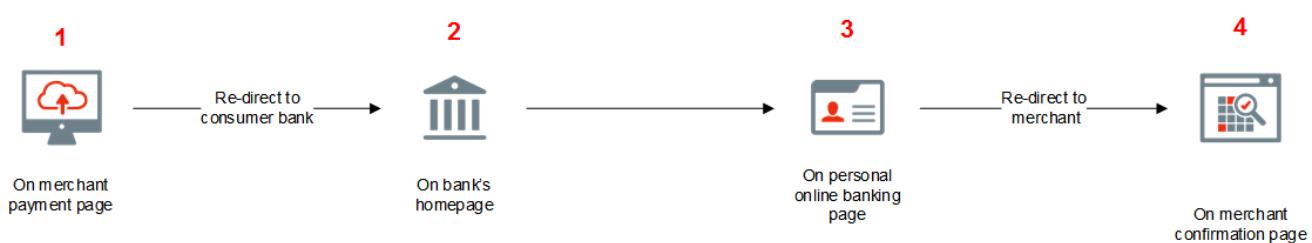
```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
<request-id>011c8c12-6340-41a2-897d-32b61f2c503b</request-id>
<transaction-type>debit</transaction-type>
<api-id>elastic-payment-page</api-id>
<payment-methods>
<payment-method name="trustpay"/>
</payment-methods>
<requested-amount currency="EUR">11.11</requested-amount>
<custom-fields><custom-field field-name="trustpay-gw-key" field-value="TestPay" /></custom-fields>
</payment>
```

A sample response for failed scenario.

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
<transaction-id>52853032-18b5-11e5-9b53-ecf4bb5fe7cb</transaction-id>
<request-id>d0408fd-a3392-409c-be60-51cc79853452</request-id>
<transaction-type>debit</transaction-type>
<transaction-state>failed</transaction-state>
<completion-time-stamp>2015-06-22T10:09:01.000+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">11.11</requested-amount>
<custom-fields>
<custom-field field-name="trustpay-gw-key" field-value="TestPay"/>
</custom-fields>
<payment-methods>
<payment-method name="trustpay"/>
</payment-methods>
<api-id>elastic-payment-page</api-id>
</payment>
```

9.9.3 Direct Banking Extension

This chapter describes direct online banking functionality of TrustPay's merchant API. This functionality allows to display bank payment options directly on his site, providing a fully customizable solution. By integrating the direct banking functionality **the customer does not have to be redirected to TrustPay's site to choose a bank**. Instead the customer is **redirected directly to the bank** which he selects from your site in case of online payments. Please note that **offline payments are not supported**.



The following elements are mandatory for sending a request for the direct banking extension for Trust Pay.

Term	Man /Opt	Format	Max Size	Description
account-holder.address.country	O	Alphanumeric	2	Country identifier of a payer.
custom-fields.custom-field field name="gw-key"	M	Alphanumeric	256	Dynamic gateway key of a bank to be returned.

Below is a GET-URL transaction with a request and response for a gateway of a specific bank to confirm debit sent to payment engine.

Request to payment engine with selected gate way of "TestPaySK"

```
POST http://127.0.0.1:8080/engine/rest/paymentmethods/?payment-method-name=trustpay HTTP/1.1
Content-Type: application/xml; charset=UTF-8
Content-Length: 552
Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.3.1 (java 1.5)
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
  <request-id>1649318a-8d44-481f-a497-41bd5477b7a7</request-id>
  <transaction-type>debit</transaction-type>
  <api-id>elastic-payment-page</api-id>
  <payment-methods>
    <payment-method name="trustpay"/>
  </payment-methods>
  <requested-amount currency="EUR">10</requested-amount>
  <custom-fields><custom-field field-name="trustpay-gw-key" field-value="TestPaySK" /></custom-fields>
</payment>
```

Response from payment engine with selected gate way of "TestPaySK"

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
  <transaction-id>887ebe5d-02b5-11e5-820d-0050b65c678c</transaction-id>
  <request-id>1649318a-8d44-481f-a497-41bd5477b7a7</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-05-25T10:10:43.000+02:00</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="EUR">10</requested-amount>
  <custom-fields>
    <custom-field field-name="trustpay-gw-key" field-value="TestPaySK"/>
  </custom-fields>
  <payment-methods>
    <payment-method url="https://ib.test.trustpay.eu/mapi/RedirectToBank.aspx?PID=4395787292&amp;URL=http%3A%2F%2F127.0.0.1%3A8080%2Fengine%2Fnotification%2Ftrustpay%2F%2Fredirect%2F&amp;NURL=https%3A%2F%2F127.0.0.1%3A8080%2Fengine%2Fnotification%2Ftrustpay%2F" name="trustpay"/>
  </payment-methods>
  <api-id>elastic-payment-page</api-id>
</payment>
```

Merchant notification for a successful debit.

```
<?xml version="1.0" encoding="UTF-8"?><payment xmlns="http://www.elastic-payments.com/schema/payment"><merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id><transaction-id>18aa5cce-02bf-11e5-820d-0050b65c678c</transaction-id><request-id>41fc0b07-45d9-46b3-9091-ced648596a06</request-id><transaction-type>debit</transaction-type><transaction-state>success</transaction-state><completion-time-stamp>2015-05-25T11:19:08.000+02:00</completion-time-stamp><statuses><status code="201.0000" description="trustpay:The resource was successfully created." severity="information"/></statuses><requested-amount currency="EUR">10</requested-amount><custom-fields><custom-field field-name="trustpay-gw-key" field-value="TestPaySK"/></custom-fields><payment-methods><payment-method name="trustpay"/></payment-methods><api-id>elastic-payment-page</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>Kfa5oTQpkAwP1xHzOopLzNDI+f8=</Digest Value></Reference></SignedInfo><SignatureValue>aS9Bz6344fgDCrLGLmoA5hh1yOrT8QmaVZlmeSw9YoGzte7j7IHi dj5O7FHa6yw3NXHsBIPqM/j6 yoAX/zpb1sbNvQ9kf1Ze0uQs9QIDM4V9hcUMuoAz0gZEnICCLGoTZe1GhOky1WijOegP+ZXA5Z00 k8IsIHSNCiBRQyj4OLoCAeSoWHkOYM39Ck7hvYW96p9J6RuvpiOHQJJ/fNYPMApu3WBLKNlnInFH vD++7WOdlqlDOJHzhwEa7/hQLhL1ZuZntC9FUmedoaAs/m9oanHOFiEc7sKuZHeud9jr59dR243h 1RWtO969c2GIYXCsDGdty+AK8alV8cPK8lnzyg==</SignatureValue><KeyInfo><X509Data><X509SubjectName>CN=M anoj Sahu,OU=Operations,O=Wirecard Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UE Payments, L=Toronto, ST=ON, C=CA</X509SubjectName><X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UE CBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXNOaWMgUGF5bwVv dHMxEzARBgNVBAstCk9wZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNjI0MDQ0 NDA5WhcNMTQwMzIwMDQ0NDA5Wjb6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBAcTB1Rvcn9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXNOaWMgUGF5bwVvudHMxEzARBgNVBAstCk9w ZXJhdGlvbnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK AoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xBozFZ3Vh3XQyy3IpluEfZz7004k4HeonfTxCN etBvJ9rgNc0Crk/euMj3pOUR9WYN2eAXC0r5pUIAZhIANsxUsaIF3JKBxf7gDAik5d8RT5HaJV 4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3lyLwhDbYlyU6j4fMKyHIIAMGzW7VgKD 2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBujabkvRWblzyQuyUyDeUb QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LaGMBAAEwDQYJKoZlhvcNAQEFBQADggEB ADgkuN/e2Ify7JXdbjNjbKbd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr mk/KqOyUL+unVcR+APjxX4KP25sdkpIgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m 02qvGoK8UA/IRbIQ6DZ9hEVK5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgU jlp9LvjVNmjzfNBBk1Si0d/rhXmMzVSKj08tp1sPRKO/sjtjZBzQajpnsZ9NFfoJNdG13AzYwDP3 x/QspK0jYn1KZw1qz524VWoQuoeR8Xj30A2jntA=</X509Certificate></X509Data></KeyInfo></Signature></payment>
```

In case the **bank** where the customer is to be redirected is **not reachable**, it is up to you what you offer to the buyer, e.g. to try to proceed payment later, or select other banks. Please note, that there is no notification sent back and the payment isn't processed.

Request to payment engine with selected gate way of "TestPaySK"

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
    <request-id>05162f33-8dc8-4679-8946-539aeb0f00d7</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="EUR">10</requested-amount>
    <custom-fields>
        <custom-field field-name="trustpay-gw-key" field-value="TestPaySK"/>
    </custom-fields>
```

```

<payment-methods>
    <payment-method name="trustpay"/>
</payment-methods>
<api-id>elastic-payment-page</api-id>
</payment>

```

Response from payment engine with selected gate way of “TestPaySK” with an error status code

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>96afe4cc-ecc1-11e4-90ec-1681e6b88ec1</merchant-account-id>
    <transaction-id>7283a22f-0457-11e5-8864-f8165476bb09</transaction-id>
    <request-id>05162f33-8dcb-4679-8946-539aeb0f00d7</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>failed</transaction-state>
    <completion-time-stamp>2015-05-27T12:02:14.000+02:00</completion-time-stamp>
    <statuses>
        <status code="400.1308" description="Gateway is not available for online payments." severity="error"/>
    </statuses>
    <requested-amount currency="EUR">10</requested-amount>
    <custom-fields>
        <custom-field field-name="trustpay-gw-key" field-value="TestPaySK"/>
    </custom-fields>
    <payment-methods>
        <payment-method name="trustpay"/>
    </payment-methods>
    <api-id>elastic-payment-page</api-id>
</payment>

```

9.9.4 Supported areas

The following is a list of **currencies** (according to ISO 4217) supported by TrustPay.

Code	ID	Name
BGN	975	Bulgarian lev
CZK	203	Czech koruna
DKK	208	Danish krone
EUR	978	Euro
GBP	826	Pound Sterling
HUF	348	Forint
NOK	578	Norwegian krone
RON	946	Romanian new leu
TRY	949	Turkish lira
USD	840	US Dollar

The following is a list of customer **countries** (according to ISO 3166-1 alpha-2) supported by TrustPay.

Code	Country
BG	Bulgaria
CZ	Czech Republic
DK	Denmark
EE	Estonia
FI	Finland
GR	Greece

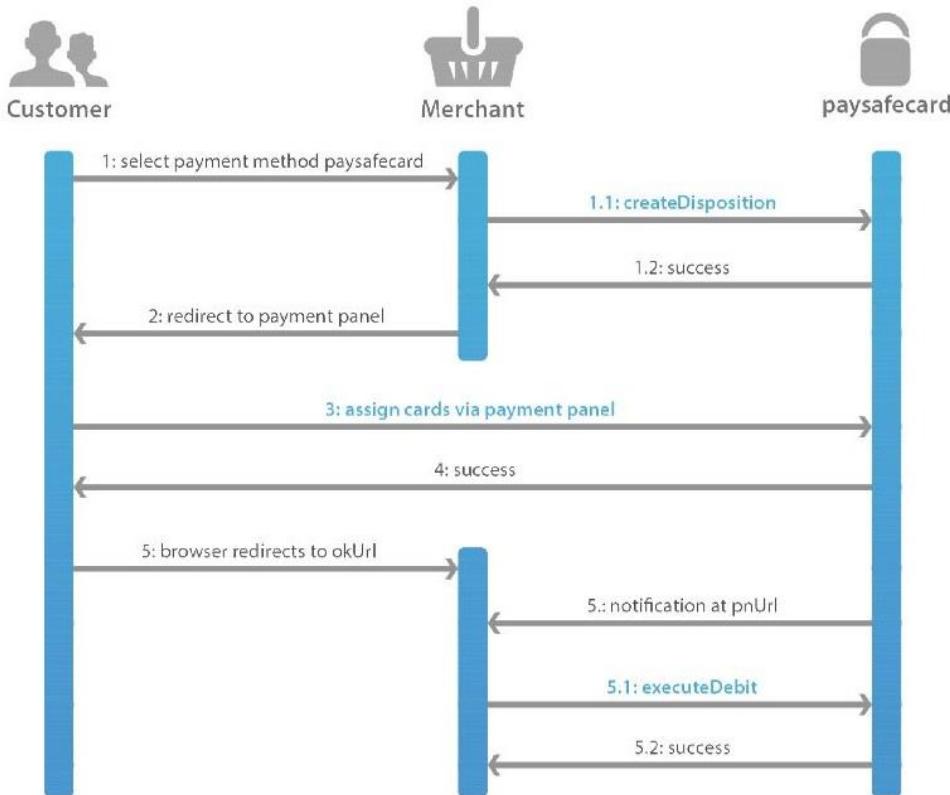
HU	Hungary
LT	Lithuania
LV	Latvia
NO	Norway
RO	Romania
SI	Slovenia
SK	Slovak Republic
TR	Turkey

If your merchant account is held in a different than the gateway's currency, you can only use such gateway if multicurrency is enabled on his account and such currency pair is supported by Trust Pay.

9.10 Paysafecard

Paysafecard is Europe's most popular and proven internet payment method: paysafecard lets you pay online simply, quickly and safely at all well-known merchants in the gaming, social media & communities and music, film & entertainment industries, as well as in many other areas.

This is a default payment flow according to paysafecard policy:



1.) Customer selects paysafecard payment method at your website

1.1) Your payment server calls *CreateDisposition* request

- If the response is NOT successful show error message to the customer.
[errorcode and resultcode = not equal 0]

1.2) Successful response from paysafecard to your payment server:

[errorcode = 0 and resultcode = 0]

2.) Redirection from website to paysafecard landing page

3.) Customer enters paysafecard PIN and clicks on pay button at paysafecard landing page

4.) Paysafecard confirms validation of the PIN

5.) pnURL -> paysafecard sends a payment notification to pnURL

5.1) Upon receipt of payment notification (pnURL) you call *executeDebit* to debit the transaction. ONLY upon successful response [errorcode = 0 and resultcode = 0] you should credit customers account

okURL -> paysafecard redirects the customer to okURL

5.2) check with *GetSerialNumbers* on okURL the state of the transaction

6.) a) if dispositionState = "S" -> call *executeDebit*

Only upon successful response [errorcode = 0 and resultcode = 0] from *executeDebit* credit customers account

b) if dispositionState = "O" -> only show success message to the customer, do not credit customers account again.

9.10.1 Fields for Paysafecard

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	18,2	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.

Additional fields which can be configured on merchant level are: country.restriction, age.limit.restriction, kyc.level.restriction.

9.10.2 XML samples

Debit request, response and notification for both success and failed scenario.

DEBIT REQUEST – SUCCESS

ENDPOINT: /engine/rest/paymentmethods

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
    <request-id>d61f59ce-aa03-4176-b9f5-44ca619d5c78</request-id>
    <transaction-type>debit</transaction-type>
    <api-id>elastic-payment-page</api-id>
    <ip-address>127.0.0.1</ip-address>
    <country>DE</country>
    <locale>DE</locale>
    <payment-methods>
        <payment-method name="paysafecard"/>
    </payment-methods>
    <requested-amount currency="EUR">1.00</requested-amount>
</payment>
```

DEBIT RESPONSE – SUCCESS (get-url transaction is created in database so far;**debit is created once the user confirms it via given payment-method url)**

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
  <transaction-id>8b1bb770-0f4c-11e5-a631-0050b667eb91</transaction-id>
  <request-id>d61f59ce-aa03-4176-b9f5-44ca619d5c78</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-10T08:41:54.000Z</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>
  <ip-address>127.0.0.1</ip-address>
  <payment-methods>
    <payment-method
      url="https://customer.test.at.paysafecard.com/psscCustomer/GetCustomerPanelServlet?currency=EUR&mtid=d61f59ce-aa03-4176-b9f544ca619d5c78&amount=1.00&mid=1000000326&locale=DE"
      name="paysafecard"/>
    </payment-methods>
    <api-id>elastic-payment-page</api-id>
    <locale>DE</locale>
    <country>DE</country>
  </payment>

```

DEBIT NOTIFICATION – SUCCESS (debit transaction is created in database; get-url transaction is parent transaction)

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64
  </merchant-account-id>
  <transaction-id>8c37eff1-0f4c-11e5-a631-0050b667eb91</transaction-id>
  <request-id>d61f59ce-aa03-4176-b9f5-44ca619d5c78</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-10T08:41:55.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000"
      description="paysafecard:The resource was successfully created."
      severity="information" />
  </statuses>
  <requested-amount currency="EUR">1.000000</requested-amount>
  <ip-address>127.0.0.1</ip-address>
  <payment-methods>
    <payment-method name="paysafecard" />
  </payment-methods>
  <api-id>elastic-payment-page</api-id>
  <country>DE</country>
</payment>

```

DEBIT REQUEST – FAILED**ENDPOINT: /engine/rest/paymentmethods**

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
    <request-id>a8cb79d6-b326-8d5e-d491-e95be3bf8569</request-id>
    <transaction-type>debit</transaction-type>
    <ip-address>127.0.0.1</ip-address>
    <country>SK</country>
    <locale>sk</locale>
    <account-holder>
        <first-name>JohnXXX</first-name>
        <last-name>Doe</last-name>
    </account-holder>
    <payment-methods>
        <payment-method name="paysafecard"/>
    </payment-methods>
    <requested-amount currency="EUR">0</requested-amount>
</payment>
```

DEBIT RESPONSE - FAILED

```
<payment
    xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
    <transaction-id>35d8a7db-0f50-11e5-a631-0050b667eb91</transaction-id>
    <request-id>a8cb79d6-b326-8d5e-d491-e95be3bf8569</request-id>
    <transaction-type>debit</transaction-type>
    <transaction-state>failed</transaction-state>
    <completion-time-stamp>2015-06-10T09:08:09.000Z</completion-time-stamp>
    <statuses>
        <status code="500.1154" description="The amount requested is not valid." severity="error"/>
    </statuses>
    <requested-amount currency="EUR">0</requested-amount>
    <account-holder>
        <first-name>JohnXXX</first-name>
        <last-name>Doe</last-name>
    </account-holder>
    <ip-address>127.0.0.1</ip-address>
    <payment-methods>
        <payment-method name="paysafecard"/>
    </payment-methods>
    <locale>sk</locale>
    <country>SK</country>
</payment>
```

CAPTURE-AUTHORIZATION REQUEST – SUCCESS**ENDPOINT: /engine/rest/payments**

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
  <request-id>0a312475-37bf-45cf-a487-b561b370ecd2</request-id>
  <transaction-type>capture-authorization</transaction-type>
  <parent-transaction-id>95059968-0f51-11e5-a631-0050b667eb91</parent-transaction-id>
  <api-id>elastic-payment-page</api-id>
  <ip-address>127.0.0.1</ip-address>
  <country>DE</country>
  <locale></locale>
  <payment-methods>
    <payment-method name="paysafecard"/>
  </payment-methods>
  <requested-amount currency="EUR">1.00</requested-amount>
</payment>
```

CAPTURE-AUTHORIZATION RESPONSE – SUCCESS

```
<payment self="http://127.0.0.1:8080/engine/rest/merchants/deb55e87-a1eb-4604-a107-dbbea003fb64/payments/959ae6e8-0f51-11e5-a631-0050b667eb91"
  xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id ref="http://127.0.0.1:8080/engine/rest/merchants/deb55e87-a1eb-4604-a107-dbbea003fb64">deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
  <transaction-id>959ae6e8-0f51-11e5-a631-0050b667eb91</transaction-id>
  <request-id>0a312475-37bf-45cf-a487-b561b370ecd2</request-id>
  <transaction-type>capture-authorization</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-10T09:17:59.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="paysafecard:The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="EUR">1.00</requested-amount>
  <ip-address>127.0.0.1</ip-address>
  <payment-methods>
    <payment-method name="paysafecard"/>
  </payment-methods>
  <api-id>elastic-api</api-id>
  <locale/>
  <country>DE</country>
</payment>
```

VOID-AUTHORIZATION REQUEST – SUCCESS**ENDPOINT: /engine/rest/payments**

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
  <request-id>3909bd28-97a0-407a-b86e-3592d94a7ef7</request-id>
  <transaction-type>void-authorization</transaction-type>
  <requested-amount currency="EUR">1.00</requested-amount>
  <parent-transaction-id>9d7b4f05-0f51-11e5-a631-0050b667eb91</parent-transaction-id>
  <payment-methods>
    <payment-method name="paysafecard"/>
  </payment-methods>
</payment>
```

VOID-AUTHORIZATION RESPONSE - SUCCESS

```

<payment self="http://127.0.0.1:8080/engine/rest/merchants/deb55e87-a1eb-4604-a107-
dbbea003fb64/payments/9def7816-0f51-11e5-a631-0050b667eb91" xmlns="http://www.elastic-
payments.com/schema/payment">
  <merchant-account-id ref="http://127.0.0.1:8080/engine/rest/merchants/deb55e87-a1eb-4604-a107-
dbbea003fb64">deb55e87-a1eb-4604-a107-dbbea003fb64</merchant-account-id>
  <transaction-id>9def7816-0f51-11e5-a631-0050b667eb91</transaction-id>
  <request-id>3909bd28-97a0-407a-b86e-3592d94a7ef7</request-id>
  <transaction-type>void-authorization</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-10T09:18:13.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="paysafecard:The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="EUR">1.00</requested-amount>
  <ip-address>127.0.0.1</ip-address>
  <payment-methods>
    <payment-method name="paysafecard"/>
  </payment-methods>
  <api-id>elastic-api</api-id>
  <country>DE</country>
</payment>
```

9.11 3pay

3pay is considered to be one of the leading payment providers in Turkey. The service spectrum ranges from mobile payment/direct carrier billing services to its own prepaid card platform.

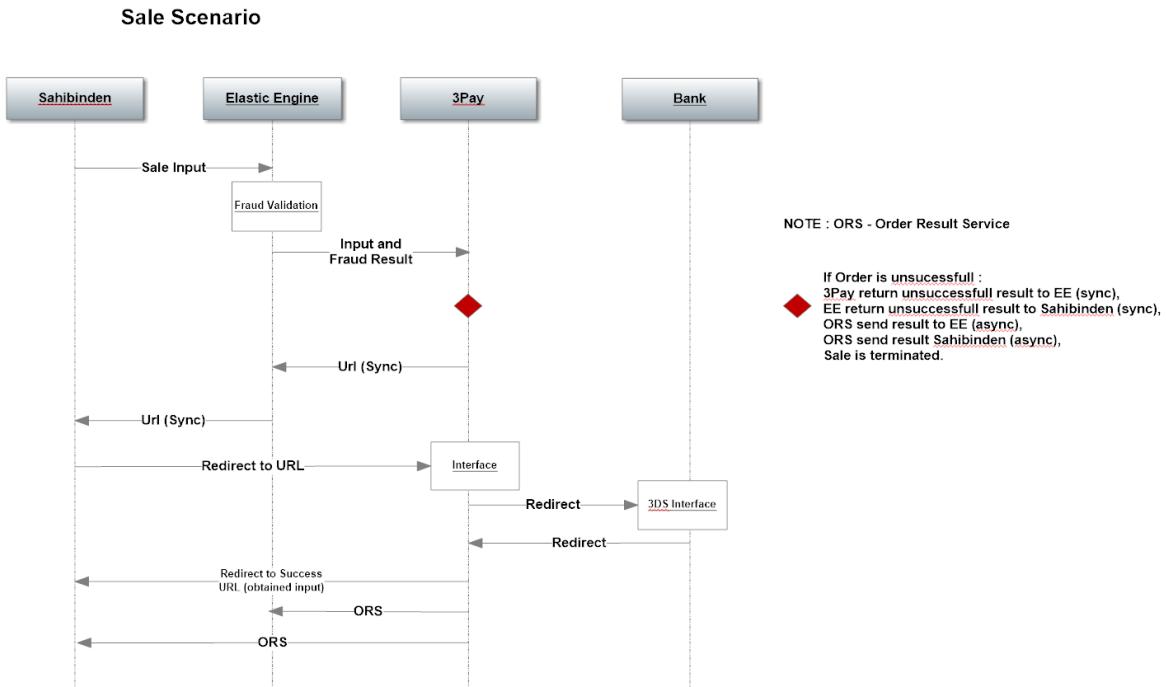
We have integrated 3Pay's "Pro API" which is a web service based solution for the payment processes that are done via 3pay|Mikro Ödeme common payment page. The solution suits the companies that make their sales at a web site and have a varying number of products/services with changing prices. Mobile payment is available as well as credit card.

In this model, the merchant company's system transmits the product price and category data to a web service hosted by 3pay|Mikro Ödeme. This web service generates a new link for every process, and returns that link to the merchant's system as response. When a customer clicks the sale link of a product at the merchant's web site, the link redirects the customer to 3pay|Mikro Ödeme common payment page. The customer enters his phone or credit card number at this common payment page, and is redirected back to the customer's web site after the payment process completes.

There are four methods supported within 3Pay and they vary in fields included in the request:

- Sale – the most basic payment method; is also a part of all following methods
- Sale3DSec – a payment with a 3D-Secure authentication
- Sale3DSECMarketplace - a payment with a 3D-Secure authentication for marketplace merchants
- SaleWithTicket (card payment) – a card payment
- SaleWithTicket (carrier-billing) – a mobile payment

9.11.1 Workflow of 3pay



9.11.2 Fields

9.11.2.1 Fields for "Sale"

Term	Man /Opt	Format	Description
merchant-account-id	M	Alphanumeric	Unique identifier of a merchant account.
request-id	M	Alphanumeric	This is the identification number of the request. It has to be unique for each request.
transaction-type	M	Numeric	This is the type for a transaction.
requested-amount currency	M	Numeric	This is the amount and currency of the transaction. The amount of the decimal place is dependent of the currency.
payment-method name	M	Numeric	The name of the Payment Method.
first-name	O	Numeric	The first name of the account holder.
last-name	M	Alphanumeric	The last name of the account holder.
email	O	Numeric	Cardholder's email.
street1	O	Numeric	Cardholder's street.
street2	O	Numeric	Cardholder's street.
city	O	Alphanumeric	Cardholder's city.
state	O	Alphanumeric	Cardholder's federation state.
country	O	Alphanumeric	Cardholder's country.
card	M	Numeric	Complete card data
ip-address	M	Numeric	Cardholder's IP-address.

9.11.2.2 Fields for "Sale 3D Sec"

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

transaction-type	M	Numeric	This is the type for a transaction.
requested-amount currency	M	Numeric	This is the amount and currency of the transaction.
			The amount of the decimal place is dependent of the currency.
payment-method name	M	Numeric	The name of the Payment Method.
first-name	O	Numeric	The first name of the account holder.
last-name	M	Alphanumeric	The last name of the account holder.
email	O	Numeric	Cardholder's email.
street1	O	Numeric	Cardholder's street.
street2	O	Numeric	Cardholder's street.
city	O	Alphanumeric	Cardholder's city.
state	O	Alphanumeric	Cardholder's federation state.
country	O	Alphanumeric	Cardholder's country.
card	M	Numeric	Card data
ip-address	M	Numeric	Cardholder's IP-address.
attempt-three-d	M	Alphanumeric	An attempt for a 3D transaction.

For method **Sale3DSec** there is <three-d> tag which checks whether the card used is 3D enrolled. At first, buyers are redirected to bank to confirm the payment and then redirected back to merchant's landing page.

9.11.2.3 Fields for "Sale 3D Sec Marketplace"

Term	Man /Opt	Format	Description
merchant-account-id	M	Alphanumeric	Unique identifier of a merchant account.
request-id	M	Alphanumeric	This is the identification number of the request. It has to be unique for each request.
transaction-type	M	Numeric	This is the type for a transaction.
requested-amount currency	M	Numeric	This is the amount and currency of the transaction.
			The amount of the decimal place is dependent of the currency.
payment-method name	M	Numeric	The name of the Payment Method.
first-name	O	Alphanumeric	The first name of the account holder.
last-name	M	Alphanumeric	The last name of the account holder.
email	O	Numeric	Cardholder's email.
street1	O	Numeric	Cardholder's street.
street2	O	Numeric	Cardholder's street.
city	O	Alphanumeric	Cardholder's city.
state	O	Alphanumeric	Cardholder's federation state.
country	O	Alphanumeric	Cardholder's country.
card	M	Numeric	Card data.
ip-address	M	Numeric	Cardholder's IP-address.
attempt-three-d	M	Alphanumeric	An attempt for a 3D transaction.
submerchant-id	M	Alphanumeric	Subpartner ID received by calling 'AddSubPartner' method. After that you can use this id to do payment. You should not send to any other ids while calling payment method, if you send, you will get error from service.
commission-rate	M	Alphanumeric	Additional info about commission rate, value can be equal or greater than zero.
installment-count	M	Alphanumeric	Installement option, can be equal zero or can be bigger than zero (Max is 9).

success-redirect-url	O	Alphanumeric	A success redirect url where a shopper is to be redirected after payment.
fail-redirect-url	O	Alphanumeric	A fail redirect url where a shopper is to be redirected after payment.
descriptor	O	Alphanumeric	Description of the product.

For method **Sale3DSECMarketplace** there is not only the <three-d> tag, but also three custom fields to identify merchant within marketplace.

9.11.2.4 Fields for “Sale with ticket”

9.11.2.4.1 Fields for “Sale with ticket” – via card payment

Term	Man /Opt	Format	Description
UserCode	M	Alphanumeric	Merchant identifier assigned by 3pay Mikro Ödeme
Pin	M	Alphanumeric	Pin value of the UserCode assigned by 3pay Mikro Ödeme
MPAY	O	Alphanumeric	An optional value assigned by the merchant's system. This can be selected as the order ID at the merchant's system or the customer's username.
Content	O	Alphanumeric	Content of an order
SendOrderResult	O	Alphanumeric	Set true/false to receive order result notifications.
PaymentTypeId	M	Numeric	Payment type (1 – 13) according to a frequency type.
ReceivedSMSObjectId	M	Numeric	Always pass '00000000-0000-0000-0000-000000000000' value.
ProductId	M	Numeric	Always pass '0' value.
ProductCategory	M	Numeric	Category of the product/service according to industry.
Price	M	Numeric	Amount of transaction.
Unit	M	Numeric	Always pass '1' value.
SendNotificationSMS	M		The notification SMS sent to the customer after a process set 'True' / 'False' for a notification SMS.
OnSuccessfulSMS	O	Alphanumeric	The message that is sent to the customer after a successful process if SendNotificationSMS parameter is passed 'true' (do not use non-standard Latin characters).
OnErrorSMS	O	Alphanumeric	The message that is sent to the customer after a failed process if SendNotificationSMS parameter is passed 'true' (do not use non-standard Latin characters).
RequestGsmOperator	M	Numeric	Pass '0' for mobile and credit card payments.
RequestGsmType	M	Numeric	Always pass '0' value.
Url	O	Alphanumeric	The web site at which the process is realized.
SuccessfulPageUrl	O	Alphanumeric	The web page address which the customer is redirected to after a successful process.
ErrorPageUrl	O	Alphanumeric	The web page address which the customer is redirected to after a failed process.
Country	O	Alphanumeric	Country info.
Currency	O	Alphanumeric	Currency info.
Extra	M	Numeric	Pass '3pay=true&cconly=true' value for credit card payment.
TurkcellServiceId	O	Numeric	The value provided by 3pay Mikro Ödeme. Must be used in every order.

When a customer clicks the sale link of a product at the merchant's web site, the link redirects the customer to Wirecard hosted payment page. The customer enters his credit card number at this common payment page, and is redirected back to the customer's web site after the payment process completes. This is a request for credit card, notice there is "<extra>3pay=true&cconly=true" tag and means this is **not** a carrier-billed transaction.

9.11.2.4.2 Fields for “Sale with ticket” – via mobile payment

Pending

9.11.3 Sample XML Request and Response for 3pay

9.11.3.1 Sample XML Request and Response for 3pay “Sale”

Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>6684b7c2-a43b-4fd4-b98c-
d606a949a3a8</merchant-account-id>
    <request-id>${unique for each request}</request-id>
    <transaction-type>purchase</transaction-type>
    <requested-amount currency="TRY">0.01</requested-amount>
    <payment-methods>
        <payment-method name="creditcard" />
    </payment-methods>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <email>john.doe@test.com</email>
        <phone/>
        <address>
            <street1>123 anystreet</street1>
            <street2/>
            <city>Brantford</city>
            <state>ON</state>
            <country>CA</country>
        </address>
    </account-holder>
    <card>
        <account-number>5157557043495547</account-number>
        <expiration-month>9</expiration-month>
        <expiration-year>2016</expiration-year>
        <card-type>visa</card-type>
        <card-security-code>139</card-security-code>
    </card>
    <ip-address>127.0.0.1</ip-address>
</payment>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment self="http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-
a43b-4fd4-b98c-d606a949a3a8/payments/f663447b-7c91-11e5-92da-
00059a3c7a00" xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id
ref="http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-a43b-4fd4-
b98c-d606a949a3a8">6684b7c2-a43b-4fd4-b98c-d606a949a3a8</merchant-
account-id>
    <transaction-id>f663447b-7c91-11e5-92da-
00059a3c7a00</transaction-id>
    <request-id>${response}</request-id>
    <transaction-type>purchase</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-10-27T11:03:26.000+01:00</completion-
time-stamp>
    <statuses>
        <status code="200.0000" description="risk:The request completed
successfully." severity="information"/>
```

```

        <status code="201.0000" description="3d-acquirer:The resource
was successfully created." severity="information"/>
    </statuses>
    <requested-amount currency="TRY">0.01</requested-amount>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <email>john.doe@test.com</email>
        <phone/>
        <address>
            <street1>123 anystreet</street1>
            <street2/>
            <city>Brantford</city>
            <state>ON</state>
            <country>CA</country>
        </address>
    </account-holder>
    <card-token>
        <token-id>5374115879995547</token-id>
        <masked-account-number>515755*****5547</masked-account-
number>
    </card-token>
    <ip-address>127.0.0.1</ip-address>
    <descriptor>demo descriptor</descriptor>
    <custom-fields>
        <custom-field field-name="elastic-api.risk.fraud-
verification.intercept" field-value="0"/>
    </custom-fields>
    <payment-methods>
        <payment-method name="creditcard"/>
    </payment-methods>
    <api-id>elastic-api</api-id>
</payment>

```

9.11.3.2 Sample XML Request and Response for 3pay “Sale 3D Sec”

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>6684b7c2-a43b-4fd4-b98c-
d606a949a3a8</merchant-account-id>
    <request-id>${unique for each request}</request-id>
    <transaction-type>purchase</transaction-type>
    <requested-amount currency="TRY">0.01</requested-amount>
    <payment-methods>
        <payment-method name="creditcard" />
    </payment-methods>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
        <email>john.doe@test.com</email>
        <phone/>
        <address>
            <street1>123 anystreet</street1>
            <street2/>
            <city>Brantford</city>
            <state>ON</state>
            <country>CA</country>
        </address>
    </account-holder>
    <card>

```

```

<account-number>5157557043495547</account-number>
<expiration-month>9</expiration-month>
<expiration-year>2016</expiration-year>
<card-type>visa</card-type>
<card-security-code>139</card-security-code>
</card>
<ip-address>127.0.0.1</ip-address>
<three-d>
    <attempt-three-d>true</attempt-three-d>
</three-d>
</payment>

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment self="http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-a43b-4fd4-b98c-d606a949a3a8/payments/0dbf6c38-7c92-11e5-92da-00059a3c7a00" xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>
        <ref>http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-a43b-4fd4-b98c-d606a949a3a8</ref>6684b7c2-a43b-4fd4-b98c-d606a949a3a8</merchant-account-id>
        <transaction-id>0dbf6c38-7c92-11e5-92da-00059a3c7a00</transaction-id>
        <request-id>${response}</request-id>
        <transaction-type>purchase</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2015-10-27T11:04:23.000+01:00</completion-time-stamp>
        <statuses>
            <status code="200.0000" description="risk:The request completed successfully." severity="information"/>
            <status code="201.0000" description="3d-acquirer:The resource was successfully created." severity="information"/>
        </statuses>
        <requested-amount currency="TRY">0.01</requested-amount>
        <account-holder>
            <first-name>John</first-name>
            <last-name>Doe</last-name>
            <email>john.doe@test.com</email>
            <phone/>
            <address>
                <street1>123 anystreet</street1>
                <street2/>
                <city>Brantford</city>
                <state>ON</state>
                <country>CA</country>
            </address>
        </account-holder>
        <card-token>
            <token-id>5374115879995547</token-id>
            <masked-account-number>515755*****5547</masked-account-number>
        </card-token>
        <ip-address>127.0.0.1</ip-address>
        <descriptor>demo descriptor</descriptor>
        <custom-fields>
            <custom-field field-name="elastic-api.risk.fraud-verification.intercept" field-value="0"/>
        </custom-fields>
        <payment-methods>
            <payment-method name="creditcard"/>
        </payment-methods>
    </payment>

```

```

</payment-methods>
<three-d>
  <acs-
    url>
  </acs-url>
</three-d>
<api-id>elastic-api</api-id>
</payment>

```

9.11.3.3 Sample XML Request and Response for 3pay “Sale 3D Sec Marketplace”

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>6684b7c2-a43b-4fd4-b98c-d606a949a3a8</merchant-account-id>
  <request-id>${unique for each request}</request-id>
  <transaction-type>purchase</transaction-type>
  <requested-amount currency="TRY">0.01</requested-amount>
  <payment-methods>
    <payment-method name="creditcard" />
  </payment-methods>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email>john.doe@test.com</email>
    <phone/>
    <address>
      <street1>123 anystreet</street1>
      <street2/>
      <city>Brantford</city>
      <state>ON</state>
      <country>CA</country>
    </address>
  </account-holder>
  <card>
    <account-number>5157557043495547</account-number>
    <expiration-month>9</expiration-month>
    <expiration-year>2016</expiration-year>
    <card-type>visa</card-type>
    <card-security-code>139</card-security-code>
  </card>
  <ip-address>127.0.0.1</ip-address>
  <three-d>
    <attempt-three-d>true</attempt-three-d>
  </three-d>
  <custom-fields>
    <custom-field field-name="submerchant-id" field-value="1" />
    <custom-field field-name="commission-rate" field-value="3" />
    <custom-field field-name="installment-count" field-value="0" />
  </custom-fields>
</payment>

```

Response

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

```

<payment self="http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-a43b-4fd4-b98c-d606a949a3a8/payments/07ab598d-7c93-11e5-92da-00059a3c7a00" xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>
        ref="http://127.0.0.1:8080/engine/rest/merchants/6684b7c2-a43b-4fd4-b98c-d606a949a3a8">6684b7c2-a43b-4fd4-b98c-d606a949a3a8</merchant-account-id>
        <transaction-id>07ab598d-7c93-11e5-92da-00059a3c7a00</transaction-id>
        <request-id>${response}</request-id>
        <transaction-type>purchase</transaction-type>
        <transaction-state>success</transaction-state>
        <completion-time-stamp>2015-10-27T11:11:05.000+01:00</completion-time-stamp>
        <statuses>
            <status code="200.0000" description="risk:The request completed successfully." severity="information"/>
            <status code="201.0000" description="3d-acquirer:The resource was successfully created." severity="information"/>
        </statuses>
        <requested-amount currency="TRY">0.01</requested-amount>
        <account-holder>
            <first-name>John</first-name>
            <last-name>Doe</last-name>
            <email>john.doe@test.com</email>
            <phone/>
            <address>
                <street1>123 anystreet</street1>
                <street2/>
                <city>Brantford</city>
                <state>ON</state>
                <country>CA</country>
            </address>
        </account-holder>
        <card-token>
            <token-id>5374115879995547</token-id>
            <masked-account-number>515755*****5547</masked-account-number>
        </card-token>
        <ip-address>127.0.0.1</ip-address>
        <descriptor>demo descriptor</descriptor>
        <custom-fields>
            <custom-field field-name="submerchant-id" field-value="1"/>
            <custom-field field-name="commission-rate" field-value="3"/>
            <custom-field field-name="installment-count" field-value="0"/>
            <custom-field field-name="elastic-api.risk.fraud-verification.intercept" field-value="0"/>
        </custom-fields>
        <payment-methods>
            <payment-method name="creditcard"/>
        </payment-methods>
        <three-d>
            <acs-url>https://test.3pay.com/proxycc/process.aspx?order=160d16b7-11e4-4986-8059-248631780785&ccId=c67f5adc-dd06-4c25-b45df22d39997f0c</acs-url>
        </three-d>
        <api-id>elastic-api</api-id>
    </payment>

```

9.11.3.4 Sample XML Request and Response for 3pay “Sale 3D Sec Marketplace”

9.11.3.4.1 Sample XML Request and Response for 3pay “Sale with ticket” card payment

Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SaleWithTicket xmlns="http://www.3pay.com/services/">
      <token>
        <UserCode>19939</UserCode>
        <Pin>BEF17E0CA4194E96A73E</Pin>
      </token>
      <input>
        <MPAY>Merchant's ID of an Order</MPAY>
        <Content>good user</Content>
        <SendOrderResult>true</SendOrderResult>
        <PaymentTypeId>1</PaymentTypeId>
        <ReceivedSMSObjectId>6955177b-d698-4afb-803d-
731281d60c31</ReceivedSMSObjectId>
        <!-- type of guid -->
        <ProductList>
          <MSaleProduct>
            <ProductId>0</ProductId>
            <ProductCategory>2</ProductCategory>
            <ProductDescription>Product
          </ProductDescription>
            <Price>0.01</Price>
            <Unit>1</Unit>
          </MSaleProduct>
        </ProductList>
        <SendNotificationSMS>false</SendNotificationSMS>
        <OnSuccessfulSMS>Payment succ!</OnSuccessfulSMS>
        <OnErrorSMS>Payment not succ!</OnErrorSMS>
        <RequestGsmOperator>0</RequestGsmOperator>
        <RequestGsmType>0</RequestGsmType>
        <Url>http://someUrl.org</Url>
        <SuccessfulPageUrl>http://localhost:8080/shop/success.ht
ml</SuccessfulPageUrl>
        <ErrorPageUrl>http://localhost:8080/shop/error.html</Err
orPageUrl>
        <Country>TR</Country>
        <Currency>TRY</Currency>
        <Extra>3pay=true&amp;cconly=true</Extra>
        <TurkcellServiceId>0</TurkcellServiceId>
      </input>
    </SaleWithTicket>
  </soap12:Body>
</soap12:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <SaleWithTicketResponse xmlns="http://www.3pay.com/services/">
      <ns2:SaleWithTicketResult>
```

```

        <TicketObject Id>c9a94813-7c91-11e5-92da-
00059a3c7a00</TicketObject Id>
        <TransactionDate>2015-10-
27T11:02:11.000+01:00</TransactionDate>
        <StatusCode>0</StatusCode>
        <RedirectUrl>
            <![CDATA[

#### 9.11.3.4.2 Sample XML Request and Response for 3pay “Sale with ticket” card payment


```

Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
    <soap12:Body>
        <SaleWithTicket xmlns="http://www.3pay.com/services/">
            <token>
                <UserCode>19939</UserCode>
                <Pin>BEF17E0CA4194E96A73E</Pin>
            </token>
            <input>
                <MPAY>Merchant's ID of an Order</MPAY>
                <Content>good user</Content>
                <SendOrderResult>true</SendOrderResult>
                <PaymentTypeId>1</PaymentTypeId>
                <ReceivedSMSObjectId>2a4e076f-d61a-4d9d-aa96-58ef7080c2cd</ReceivedSMSObjectId>
                <!-- type of guid -->
                <ProductList>
                    <MSaleProduct>
                        <ProductId>0</ProductId>
                        <ProductCategory>2</ProductCategory>
                        <ProductDescription>Product      that      kicks
ass</ProductDescription>
                        <Price>0.01</Price>
                        <Unit>1</Unit>
                    </MSaleProduct>
                </ProductList>
                <SendNotificationSMS>false</SendNotificationSMS>
                <OnSuccessfulSMS>Payment succ!</OnSuccessfulSMS>
                <OnErrorSMS>Payment not succ!</OnErrorSMS>
                <RequestGsmOperator>0</RequestGsmOperator>
                <RequestGsmType>0</RequestGsmType>
                <Url>http://someUrl.org</Url>
            </input>
        </SaleWithTicket>
    </soap12:Body>
</soap12:Envelope>
```

```

<SuccessfulPageUrl>http://www.google.com/</SuccessfulPageUrl>
<ErrorPageUrl>http://www.ubuntu.com/</ErrorPageUrl>
<Country>TR</Country>
<Currency>TRY</Currency>
<TurkcellServiceId>0</TurkcellServiceId>
</input>
</SaleWithTicket>
</soap12:Body>
</soap12:Envelope>

```

Response

1f32df99-f06f-4770-a072-022e85131d41 2015-10-27T11:00:09.766+01:00 0
--

9.12 Alipay Cross Border

Alipay Cross-Border E-Payment Service is a payment solution that allows buyers to pay for goods sold on international partner merchant websites using Renminbi (RMB). Alipay will then remit the sum in a foreign currency (one of the twelve that are currently supported) to the international merchant in settlement. The end consumer is redirected to Alipay and authorizes the transaction. A notification confirming the outcome of the transaction is sent to the merchant. The notification for the final status of a payment should be received within a few minutes. If the notification is not received within this time, the merchant has the possibility to send a request for “Retrieve Transaction by Transaction ID” or “Retrieve Transaction by Request ID”. With the response the merchant will get the current status of the requested transaction.

9.12.1 Fields

The following elements are mandatory/optional:

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 birth date.
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
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
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 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 is generated by Wirecard.
transaction-type	M	Alphanumeric	30	This is the type for a transaction. For Alipay Cross-border Debit and Debit-Refund is allowed.
payment-methods/payment-method/@url	M	Alphanumeric	2056	This is the URL which the consumer must be redirected to in order to complete the payment.

9.12.2 XML Samples for Alipay Cross Border

9.12.2.1 Debit

Successful Debit Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <payment-methods>
        <payment-method name="alipay-xborder"/>
    </payment-methods>
    <merchant-account-id>pending</merchant-account-id>
    <request-id>${unique for each request}</request-id>
    <transaction-type>debit</transaction-type>
    <requested-amount currency="USD">1.01</requested-amount>
    <order-number>37530</order-number>
    <order-detail>Test product 001</order-detail>
    <ip-address>127.0.0.1</ip-address>
```

```

<locale>en</locale>
<account-holder>
  <first-name>Max</first-name>
  <last-name>Cavalera</last-name>
  <email>max.cavalera@wirecard.com</email>
</account-holder>
</payment>

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>pending</merchant-account-id>
  <transaction-id>6c08d5b2-0f85-11e5-87be-00163e5411b5</transaction-id>
  <request-id>${response}</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-10T15:29:02.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="USD">1.01</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email>john.doe@wirecard.com</email>
  </account-holder>
  <ip-address>127.0.0.1</ip-address>
  <order-number>37530</order-number>
  <order-detail>Test product 001</order-detail>
  <payment-methods>
    <payment-method
      url="https://mapi.alipay.net/gateway.do?sign type=MD5&partner=2088101122136241&service=create forex trade&notify url=https%3A%2F%2Fsandbox-engine.thesolution.com%2Fengine%2Fnofication%2Falipay-xborder%2F&return url=https%3A%2F%2Fsandbox-engine.thesolution.com%2Fengine%2Fnofication%2Falipay-xborder%2Fredirect%2F6c08d5b2-0f85-11e5-87be-00163e5411b5%2F&out trade no=6c08d5b2-0f85-11e5-87be-00163e5411b5&subject=37530&total fee=1.01&currency=USD&sign=7526821a09913332320a161ba161206b" name="alipay-xborder"/>
  </payment-methods>
  <locale>en</locale>
</payment>

```

Notification

```

<?xml version="1.0" encoding="UTF-8"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>pending</merchant-account-id>
  <transaction-id>f35a71bf-16bc-11e5-87be-00163e5411b5</transaction-id>
  <request-id>${response}</request-id>
  <transaction-type>debit</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-19T19:54:09.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="alipay-xborder:The resource was successfully created." severity="information" />
  </statuses>
</payment>

```

```

<requested-amount currency="USD">1.01000</requested-amount>
<account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email>alipaytest20091@gmail.com</email>
    <phone>5555555555</phone>
    <address>
        <street1>150 Eglinton</street1>
        <street2>Suite 111</street2>
        <city>Toronto</city>
        <state>ON</state>
        <country>CA</country>
        <postal-code>M4P1E8</postal-code>
    </address>
</account-holder>
<ip-address>127.0.0.1</ip-address>
<order-number>PO98U457Z</order-number>
<order-detail>1 XL TShirt (white)</order-detail>
<notifications>
    <notification url="http://requestb.in/11h0jull1" />
</notifications>
<custom-fields>
    <custom-field field-name="Invoice Number" field-value="IN4523H11" />
</custom-fields>
<payment-methods>
    <payment-method name="alipay-xborder" />
</payment-methods>
<api-id>elastic-payment-page</api-id>
<processing-redirect-url />
<cancel-redirect-url />
<fail-redirect-url />
<success-redirect-url />
<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" />
            </Reference>
        </SignedInfo>
        <SignatureValue>GCBqKY9QnrlnZ3AG/iFqQinZh1LJXQefLVvSAyKsyXxQoO2xVvj1fsgs17CjJ6HJFr/GNGEsfrYyA627wftih8u0cux+uAg5a+tuOTEgIbXdkbh2sUWaDPUnHpaHqsrt+C59TD5kpNa6ABmaUvo2EcLoARzN1HBv65Zo05sTvn/X1EmKjZ0D3L/XzgBuZngfAFLnu+/Wtny021kEpDSRB/Bh14RNxp45fji4oAzztg9/DNV DipiE5medvNkkrdhcxJ2XU11PT2q8fJyY4YsFrzBDTe sODIVailLzbMtE+yfiEu+gWzWOPBi5FW83h06ueXvY1BjKNEbKpoGU7hifkrw==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509SubjectName>CN=*.thesolution.com, OU=Domain Control Validated, O=*.thesolution.com</X509SubjectName>
                <X509Certificate>MIIFatCCBFggAwIBAgIHS05K//8WnzANBgkqhkiG9w0BAQUFADCByjELMAkGA1UEBhMCVVmxEDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTC1Njb3R0c2RhbgUxGjAYBgNVBAoTEUdvRGFkZHkuY29tLCBJbmMuMTMwMQYDVQQLEypodHRwOi8vY2VydGlmaWNhdGVzLmdvZGFkZHkuY29tL3JlcG9zaXRv
    </X509Data>
</X509SubjectName>
<X509Certificate>MIIFatCCBFggAwIBAgIHS05K//8WnzANBgkqhkiG9w0BAQUFADCByjELMAkGA1UEBhMCVVmxEDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTC1Njb3R0c2RhbgUxGjAYBgNVBAoTEUdvRGFkZHkuY29tLCBJbmMuMTMwMQYDVQQLEypodHRwOi8vY2VydGlmaWNhdGVzLmdvZGFkZHkuY29tL3JlcG9zaXRv

```

```

cnkxMDAuBgnVBAMTJ0dvIERhzGR5IFN1Y3VyzSBDZXJ0aWzpy2F0aW9uIEF1dGhvcml0eTERMA8G
A1UEBRMIMDC5NjkyODcwHhcNMTExMTAxMTM1MzQxWhcNMТИxMjAxMTY0MTUwWjBbMRowGAYDVQQK
DBEqLnRoZXNvbHV0aW9uLmNvbTEhMB8GA1UECwWYRG9tYWluIEvbnRyb2wgVmFsWRhdGVkMRow
GAYDVQQDBEqLnRoZXNvbHV0aW9uLmNvbTCCASIWdQYJKoZIhvcNAQEBQADggEPADCCAQoCggEB
AKdw14rJ3PjXr8NbZFypT14XRkpBLuwH+Y8+GPq7mb+X4D20ZhwQk1PvO1DsHbnKf4Pfbx1SeX+i
jwJ78fDuM9yezS193wzmzJzBnpUkvduIPD7lBpx87tIVT9Hcpt9dMrqF4Y9UByzOws77gMux5gHF
tV9KmSL7RF0rHSCm5/27n9hzxAffspZxhS311cBKgkIi1tt501GnCGCzvFruyw2R7W/Z9MxxtLoN
+YdB6i04fc4oygtvzt4DycZ0zwjT10W3E7/k/xGUx/pYH9RLN3Ni9xanRLESJ1t2ohFXoE2eArP
+y+QAyj8c4P+k6bPwZegXCFsnHXP6VJyDZtOLSECAwEAAAOCAcAwggG8MA8GA1UdEwEB/wQFMAMB
AQAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMA4GA1UdDwEB/wQEAvIFoDAzBgNVHR8E
LDAqMCigJqAkhiJodHRwOi8vY3JsLmdvZGFkZHkuY29tL2dkczEtNTkuY3JsMFMGAA1UDiARMMEow
SAYLYIZIAyb9bQEHFWewOTA3BggRgEFBQcCARYraHR0cDovL2N1cnRpZmljYXRlcj5nb2RhZGR5
LmNvbS9yZXBvc210b3J5LzCBgAYIKwYBBQUHAQEEdDByMCQGCCsGAQUFBzABhhodHRwOi8vb2Nz
cC5nb2RhZGR5LmNvbS8wSgYIKwYBBQUHMAKGPMh0dHA6Ly9jZXJ0aWZpY2F0ZXMuZ29kyWRkeS5j
b20vcmVwb3NpdG9ye9nZF9pbnR1cm1lZG1hdGUuY3J0MB8GA1UdIwQYMBAFP2sYTKTbEXW4u6F
X5q653aZaMznMC0GA1UdEQQmMCSCESoudGh1c29sdXRpb24uY29tgg90aGVzb2x1dGlvb15jb20w
HQYDVR0OBByEFGwt8Z80Urg6pSHSiut3idSbCOHMA0GCSqGSIb3DQEBBQUAA4IBAQBrdO6615sg
s9zUIKiY1oQsKah69RkcQo1f7cwBwNDQLi/DY0KD4efWjOjeon3noaoJ26rrfKyfCE8ShmjGcOuL
4YdEFdvuFgeIBe86jg5CnVF0vNfdiaIMJXAE/SaoK/b/Nabepcvwc2/A+snNaf4kmJkDbHg3IVIY
62znQhzyQWzpYRYaA36X1BxhoYoz8jKLntfSF/SvXCy/f+J9+55T1r/U2kM5Ulem7z9PsajUHJj1
M35h6Phh1kFU8r0FCuySsfPdyqZe5Xc/bP7yOub4XINvnIEZMcDQfL+PCIEdYDxNbYM6+gtCl78
8Bz9XY5A6TrXcazIPwW/t3jTZRs/</X509Certificate>
    </X509Data>
</KeyInfo>
</Signature>
</payment>

```

9.12.2.2 Refund Debit

Successful Refund Debit Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>pending</merchant-account-id>
    <request-id>88bba544-5d77-4f52-9b43-531642b2543f</request-id>
    <transaction-type>refund-debit</transaction-type>
    <parent-transaction-id>${former debit transaction}</parent-transaction-id>
    <ip-address>127.0.0.1</ip-address>
    <payment-methods>
        <payment-method name="alipay-xborder"/>
    </payment-methods>
</payment>

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>pending</merchant-account-id>
    <transaction-id>a855aad4-b62f-11e5-aecf-00163e0e673e</transaction-id>
    <request-id>88bba544-5d77-4f52-9b43-531642b2543f</request-id>
    <transaction-type>refund-debit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2016-01-08T17:45:52.000Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="alipay-xborder:The resource was successfully created." severity="information"/>
    </statuses>
    <requested-amount currency="USD">1.01</requested-amount>

```

```

<account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
    <email>john.doe@wirecard.com</email>
    <phone>1 555 555 5555</phone>
    <address>
        <street1>123 test</street1>
        <city>Toronto</city>
        <state>ON</state>
        <country>CA</country>
        <postal-code>M4P1E8</postal-code>
    </address>
</account-holder>
<ip-address>127.0.0.1</ip-address>
<order-number>123456</order-number>
<order-detail>1 widget</order-detail>
<payment-methods>
    <payment-method name="alipay-xborder"/>
</payment-methods>
<api-id>elastic-api</api-id>
<processing-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp?state=processing&</processing-redirect-url>
    <cancel-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp?state=cancel&</cancel-redirect-url>
    <fail-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp?state=failed&</fail-redirect-url>
    <success-redirect-url>https://sandbox-engine.thesolution.com/shop/complete.jsp?state=success&</success-redirect-url>
<locale>en</locale>
</payment>

```

10 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.

10.1 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.

10.2 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

10.2.1 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 6.4).

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.

10.2.1.1 Fields

The following elements are mandatory (X) / optional (O) for sending a request/response for the payment method **SEPA Direct Debit**.

Field	Request	Response	Notification	Datatype	Size	Description
merchant-account-id	X		X	String	36	Unique identifier for a merchant account.
transaction-id		X	X	String	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	X			String	150	This is the identification number of the request on the merchant's side. It must be unique for each request.
transaction-type	X			String	30	This is the type for a transaction. For SEPA Direct Debit, only <i>DEBIT</i> is allowed.
transaction-state		X	X	String	12	This is the status of a transaction
completion-time-stamp		X	X	dateTime		This is the timestamp of completion of request.
status.code		X	X	String	12	This is the code of the status of a transaction.
status.description		X	X	String	256	This is the description to the status code of a transaction.
status.severity		X	X	String	20	This field gives information if a status is a warning, an error or an information.
statuses.Status		X	X	String	12	This is the status of a transaction.
requested-amount	X	X	X	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-amount@currency	X	X	X	String	3	This is the currency of the requested-amount. Only EUR (euro) is accepted
parent-transaction-id	O	O	O	String	36	Transaction ID of the first transaction of a payment.
account-holder.first-Name	X	X	X	String	32	This is the first name of the end-consumer.

account-holder.last-name	X	X	X	String	32	This is the last name of the end-consumer.
account-holder.email	O	O	O	String	64	This is the end-consumer's Email-Address.
account-holder.gender	O	O	O	String	1	This is the end-consumer's gender.
account-holder.date-of-birth	O	O	O	Date		This is the end-consumer's birth date.
account-holder.phone	O	O	O	String	32	This is the phone number of the end-consumer.
account-holder.address.street1	O	O	O	String	70	This is the first part of the end-consumer's street.
account-holder.address.street2	O	O	O	String	128	This is the second part of the end-consumer's street.
account-holder.address.city	O	O	O	String	32	This is the end-consumer's city.
account-holder.address.state	O	O	O	String	32	This is the end-consumer's state.
account-holder.address.country	O	O	O	String	3	This is the end-consumer's country.
account-holder.address.postal-code	O	O	O	String	16	This is the end-consumer's postal code.
ip-address	O	O	O	String	15	The global (internet) IP address of the consumer's computer.
order-number	O	O	O	String	64	This is the order number of the merchant.
order-detail	O	O	O	String	65535	This is a field for details of an order filled by the merchant.
descriptor	O	O	O	String	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.
notifications.notification@url	O	O	O	String	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.

custom-field.field-name	O	O	O	String	36	This is the name for the custom field.
custom-field.field-value	O	O	O	String	256	This is the content of the custom field. In this field the merchant can send additional information.
payment-methods.payment-method-Name	X	X	X	api-id	15	This is the name of the payment method ,sepadirectdebit'
api-id			X	api-id		The API id is always returned in the notification. For SEPA it is "___"
bank-account.bic	O	O	O	String	8 or 11	<p>This is the Business Identifier Code of the bank of the end-consumer.</p> <p>Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z0-9]{2}([a-zA-Z0-9]{3})</p>
bank-account.iban	X/O	X/O	X/O	String	34	<p>This is the International Bank Account Number of the end-consumer.</p> <p>Allowed characters: [a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}</p>
mandate.mandate-id	X	X	X	String	35	<p>The ID of the signed mandate between the merchant and the consumer.</p> <p>The mandate ID may NOT be generated by the Wirecard Payments Platform. It is solely the responsibility of the merchant.</p> <p>Allowed characters: [A-Za-z0-9][\+ \? / - : \\() \\. , '){1,35}</p>
mandate.signature-date	X	X	X	Date	n/a	This is the date on which the above-mentioned mandate

						was signed by the consumer.
						The date cannot be in the future, the validity is checked against the server time. Merchant may choose to specify the UTC timezone as +/- number of hours, e.g. <signed-date>2013-09-24+3</signed-date>. The timezone is considered during the validation process, sign date is stored with transaction using server's timezone afterwards.
						The Mandate Signature Date is only required for SEPA Direct Debit and not for SEPA Credit Transfer transactions.
creditor-id	X	X	X	String	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}
due-date	O	O	O	Date	O	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.
provider-transaction-reference-id		X	X	String	10	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.
--	--	--	--	--	---

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¬First=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

10.2.1.2 Delayed processing of SEPA direct debit transactions

It is possible to delay the processing of a direct debit SEPA transaction by providing the element payment/capture-date inside the SEPA direct debit request. Only on the day provided with this element the corresponding SEPA transaction will be sent to the Wirecard bank for the processing. Thus it is possible to cancel this transaction before the cut-off time of the capture date. The capture date can be in range of the following day and the following + 14 calendar days. The capture date out of this range will lead to a transaction rejection with the status code 400.1321. If the capture date is a bank non-working day, the processing will start on the next bank working day. For example, if the capture date is pointing to a Saturday, the transaction processing will start on next Monday.

The capture date, if provided, influences the validation rule and the calculation logic of the due date. The due date is validated and calculated according to the same rules as described in the chapter 'Due Date' but starting from the capture date. For example, if the capture date is the Tuesday 25.11.2015, the next possible due date for the one-off transaction is 25.11.2015 + 7 bank working days = Friday 04.12.2015.

The due date may only be used if it passes validation. If validation is not passed, Wirecard will return a failed transaction (400.1175) and the transaction must be re-submitted.

PLEASE NOTE: the provided capture date does not guarantee the money flow on this date. The real money flow will occur according to the provided or calculated due date.

10.2.1.3 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>DE98ZZZ099999999999</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>DE98ZZZ099999999999</creditor-id>
</payment>
```

```
<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>WIREDEMXX</bic>
    </bank-account>
    <mandate>
        <mandate-id>12345678</mandate-id>
        <signed-date>2013-09-24</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ099999999999</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>WIREDEMXX</bic>
    </bank-account>
    <mandate>
        <mandate-id>12345678</mandate-id>
        <signed-date>2013-09-24</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ099999999999</creditor-id>
</payment>
```

10.2.2 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

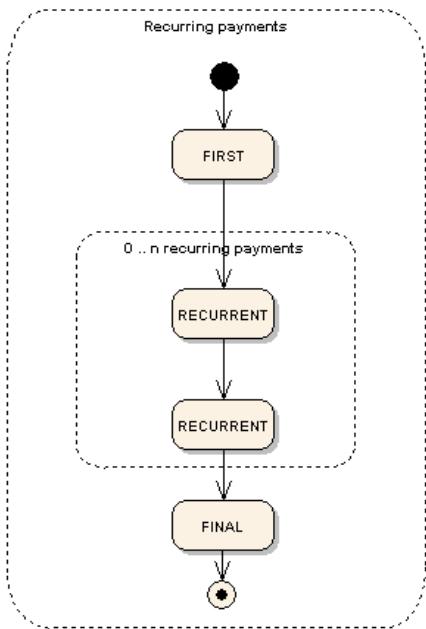
The Due Date calculation of a recurring transaction depends on the sequence type of that transaction. Please refer to section 7.2.1.1 which defines how to calculate a due date for each sequence type.

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:
 - o bankaccount
 - o paymentmethods
 - o shipping
 - o mandate
 - o airline industry
 - o 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.

10.2.2.1 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'

10.2.2.2 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>DE4251230800000060004</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>WIREDDEMMXXX</bic>
    </bank-account>
    <mandate>
        <mandate-id>12345678</mandate-id>
        <signed-date>2013-12-19</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ099999999999</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-69db83fb555</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-69db83fb555</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>DE98ZZZ099999999999</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>DE98ZZZ099999999999</creditor-id>
    <periodic>
        <periodic-type>recurring</periodic-type>

```

```
<sequence-type>recurring</sequence-type>
</periodic>
</payment>
```

10.2.3 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>DE4251230800000060004</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.

10.2.3.1 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 it will be validated to the formal correctness. Allowed characters: ([a-zA-Z]{4}[a-zA-Z]{2}[a-zA-Z]{2}[a-zA-Z]{2})?)
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.

10.2.3.2 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>WIREDEMXX</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>WIREDEMXXX</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>WIREDEMXXX</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>WIREDEMXXX</bic>
  </bank-account>

```

```
<provider-transaction-reference-id>68E34C9581</provider-transaction-reference-id>
</payment>
```

10.2.3.3 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>wgeffeosddhse</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>GENODEF10GK</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>DE98ZZZ099999999999</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>wgeffeosddhse</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>
```

10.2.4 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 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 6.4).

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.

10.2.4.1 Fields

The following elements are mandatory (X) / optional (O) for sending a request/response for the payment method **SEPA Credit Transfer**.

merchant-account-id	X		X	String	36	Unique identifier for a merchant account.
transaction-id		X	X	String	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id	X			String	150	This is the identification number of the request on the merchant's side. It must be unique for each request.
transaction-type	X			String	30	This is the type for a transaction. For SEPA Credit Transfer, only CREDIT is allowed.
transaction-state		X	X	String	12	This is the status of a transaction
completion-time-stamp		X	X	dateTime		This is the timestamp of completion of request.
status.code		X	X	String	12	This is the code of the status of a transaction.
status.description		X	X	String	256	This is the description to the status code of a transaction.
status.severity		X	X	String	20	This field gives information if a status is a warning, an error or an information.
statuses.status		X	X	String	12	This is the status of a transaction.
requested-amount	X	X	X	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-amount@currency	X	X	X	String	3	This is the currency of the requested-amount. Only EUR (euro) is accepted
parent-transaction-id	O	O	O	String	36	Transaction ID of the first transaction of a payment.
account-holder.first-Name	X	X	X	String	32	This is the first name of the end-consumer.
account-holder.last-Name	X	X	X	String	32	This is the last name of the end-consumer.

account-holder.email	O	O	O	String	64	This is the end-consumer's Email-Address.
account-holder.gender	O	O	O	String	1	This is the end-consumer's gender.
account-holder.date-of-birth	O	O	O	Date		This is the end-consumer's birth date.
account-holder.phone	O	O	O	String	32	This is the phone number of the end-consumer.
account-holder.address.street1	O	O	O	String	70	This is the first part of the end-consumer's street.
account-holder.address.street2	O	O	O	String	128	This is the second part of the end-consumer's street.
account-holder.address.city	O	O	O	String	32	This is the end-consumer's city.
account-holder.address.state	O	O	O	String	32	This is the end-consumer's state.
account-holder.address.country	O	O	O	String	3	This is the end-consumer's country.
account-holder.address.postal-code	O	O	O	String	16	This is the end-consumer's postal code.
ip-address	O	O	O	String	15	The global (internet) IP address of the consumer's computer.
order-number	O	O	O	String	64	This is the order number of the merchant.
order-detail	O	O	O	String	65535	This is a field for details of an order filled by the merchant.
descriptor	O	O	O	String	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.
notifications.notification@url	O	O	O	String	256	The URL to be used for the Instant Payment Notification. It overwrites the notification URL that is set up in the merchant configuration.
custom-field.field-name	O	O	O	String	36	This is the name for the custom field.
custom-field.field-value	O	O	O	String	256	This is the content of the custom field. In this field the merchant can send additional information.
payment-methods.payment-method@name	X	X	X	api-id	15	This is the name of the payment method ,sepacredit'
api-id			X	api-id		The API id is always returned in the notification. For SEPA it is "---"
bank-account.bic	O	O	O	String	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	X/O	X/O	X/O	String	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	O	O	O	String	1,35	Identifies and allows a merchant to process SEPA Credit Transfertransactions. Allowed characters: [a-zA-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{3,3}[a-zA-Z0-9]{1,28}
mandate.mandate-id	O	O	O	String		Every mandate has an ID provided by the merchant that, when combined with the Creditor ID, creates a unique mandate ID.
mandate.signed-date	O	O	O	String		This is the date on which the mandate was signed by the consumer.
provider-transaction-reference-id		X	X	String	10	This ID provides a reference for the complete end-to-end lifecycle of a SEPA Credit Transfertransaction. It is used as a reference within the banking system to ensure all transactions referencing each other (eg: a Credit and a debit return) are matched and that the complete lifecycle of a payment is identifiable. Wirecard generates this ID for the merchant.

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.

10.2.4.2 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>DE4251230800000060004</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>DE4251230800000060004</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>
```

10.2.5 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.

10.2.5.1 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.

10.2.5.2 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>

```

10.2.6 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.

10.2.6.1

The following elements are submitted in the IPN (Instant Payment Notification) for a **SEPA deposit**.

Field	Notification	Datatype	Size	Description
merchant-account-id	X	String	36	Unique identifier for a merchant account.
transaction-id	X	String	36	The Transaction ID is the unique identifier for a transaction. It is generated by Wirecard.
request-id		String	150	This is the identification number of the request on the merchant's side. It must be unique for each request .
transaction-type		String	30	This is the type for a transaction. For SEPA Deposit, it is always "deposit"
transaction-state	X	String	12	This is the status of a transaction
completion-time-stamp	X	dateTime		This is the timestamp of completion of request.
status.code	X	String	12	This is the code of the status of a transaction.
status.description	X	String	256	This is the description to the status code of a transaction.
status.severity	X	String	20	This field gives information if a status is a warning, an error or an information.
statuses.status	X	String	12	This is the status of a transaction.
requested-amount	X	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-amount@currency	X	String	3	This is the currency of the requested-amount. Only EUR (euro) is possible
descriptor	O	String	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.
payment-methods.payment-method@name	X	api-id	15	This is the name of the payment method , wiretransfer '
provider-transaction-reference-id	X	String	10	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.
signature	X	complex		This is the signature of the Transaction, consisting of SignedInfo, SignatureValue and KeyInfo

10.2.7 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.

10.2.7.1 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

10.2.7.2 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

10.2.7.3 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: **DE112222222333333333**
Account Holder Bank Account: **DE223333334444444444**
BIC NOT REQUIRED

Merchant Bank Account: **DE112222222333333333**
Account Holder Bank Account: **AT556666677777777777**
BIC REQUIRED

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

10.2.8 Cross-Payment-Method Referencing

The essential setting of the cross-payment-method referencing is a merchant has more alternative payment methods configured within MAID.

So far, failed SEPA transactions due to "bank account not existing" or "invalid bank data" led to debit returns. From now on it's possible to combine online banking payment methods Sofortbanking, iDEAL, and giropay with referenced SEPA transactions to reduce debit-returns and as well to improve customer conversion. The solution is available for Wirecard's Payment Processing Gateway backend and HPP/EPP.

The initial transaction is placed by the customer actively via one of the supported alternative payment methods. In case of a successful transaction the bank details IBAN & BIC are saved in the database, but not returned back in the response. The merchant instead receives a transaction ID which is used later to place referenced SEPA debit transactions without knowing the bank account details as they are automatically connected to the transaction ID. Based on this e.g. subscriptions can be managed.

All mandatory SEPA fields like mandate ID, mandate signature date and creditor ID are **not** placed in the first referenced SEPA transaction as they have been placed in the previous Sofortbanking, iDEAL, or giropay transaction. On the other hand this transaction is flagged as “first” to avoid that a “recurring” flagged SEPA transaction is rejected by the system.

It is also possible to place a SEPA credit transaction using the transaction ID of one of the supported alternative payment methods.

10.2.8.1 XML Samples

iDeal initial payment request:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <request-id>b2ddf106-0f00-41da-b8af-64429b3ee329</request-id>
    <transaction-type>get-url</transaction-type>
    <payment-methods>
        <payment-method name="ideal"/>
    </payment-methods>
    <requested-amount currency="EUR">2.31</requested-amount>
    <order-description>test order</order-description>
    <descriptor>customerStatement 18009998888</descriptor>
    <bank-account>
        <bic>INGBNL2A</bic>
    </bank-account>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
    </account-holder>
    <order-number>43606</order-number>
    <mandate>
        <mandate-id>manid</mandate-id>
        <signed-date>2015-02-02</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ09999999999</creditor-id>
    <periodic>
        <periodic-type>installment</periodic-type>
        <sequence-type>first</sequence-type>
    </periodic>
</payment>
```

iDeal initial payment response:

```
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <transaction-id>b8b27f18-4003-11e5-919b-005056a64448</transaction-id>
    <request-id>b2ddf106-0f00-41da-b8af-64429b3ee329</request-id>
    <transaction-type>get-url</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-08-11T07:07:08.000Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully created." severity="information"/>
    </statuses>
    <requested-amount currency="EUR">2.31</requested-amount>
    <account-holder>
        <first-name>John</first-name>
    </account-holder>
```

```

<last-name>Doe</last-name>
</account-holder>
<order-number>43606</order-number>
<descriptor>customerStatement 18009998888</descriptor>
<payment-methods>
    <payment-method url="https://idealtest.secure-
ing.com/ideal/issuerSim.do?trxd=0050000102513955&ideal=prob" name="ideal"/>
    </payment-methods>
    <bank-account>
        <bic>INGBNL2A</bic>
    </bank-account>
    <mandate>
        <mandate-id>manid</mandate-id>
        <signed-date>2015-02-02</signed-date>
    </mandate>
    <creditor-id>DE98ZZ0999999999</creditor-id>
    <periodic>
        <periodic-type>installment</periodic-type>
        <sequence-type>first</sequence-type>
    </periodic>
</payment>
```

Note, that the referenced SEPA payment doesn't contain mandatory fields in request and these are taken from the original iDeal payment based on parent transaction id.

Following SEPA Direct Debit payment request:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <request-id>72ae0263-384f-41cc-8caa-aa78330afca3</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>
    </account-holder>
    <payment-methods>
        <payment-method name="sepadirectdebit"/>
    </payment-methods>
    <consumer>
        <first-name/>
        <last-name/>
    </consumer>
    <periodic>
        <periodic-type>recurring</periodic-type>
        <sequence-type>recurring</sequence-type>
    </periodic>
    <parent-transaction-id>b8b27f18-4003-11e5-919b-005056a64448</parent-transaction-id>
</payment>
```

Following SEPA Direct Debit response:

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <transaction-id>bcd853f6-4003-11e5-919b-005056a64448</transaction-id>
    <request-id>72ae0263-384f-41cc-8caa-aa78330afca3</request-id>
```

```

<transaction-type>debit</transaction-type>
<transaction-state>success</transaction-state>
<completion-time-stamp>2015-08-11T08:34:10.000Z</completion-time-stamp>
<statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
</statuses>
<requested-amount currency="EUR">1.01</requested-amount>
<parent-transaction-id>b8b27f18-4003-11e5-919b-005056a64448</parent-transaction-id>
<account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
</account-holder>
<order-number>39849</order-number>
<descriptor>customerStatement 18009998888</descriptor>
<payment-methods>
    <payment-method name="sepadirectdebit"/>
</payment-methods>
<bank-account>
    <iban>NL53INGB0654422370</iban>
    <bic>INGBNL2A</bic>
</bank-account>
<mandate>
    <mandate-id>manid</mandate-id>
    <signed-date>2015-02-02</signed-date>
</mandate>
<creditor-id>DE98ZZZ0999999999</creditor-id>
<due-date>2015-08-17</due-date>
<periodic>
    <periodic-type>recurring</periodic-type>
    <sequence-type>recurring</sequence-type>
</periodic>
<consumer>
    <first-name/>
    <last-name/>
</consumer>
<provider-transaction-reference-id>261AD6D93C</provider-transaction-reference-id>
<instrument-country>NL</instrument-country>
</payment>

```

This is the scenario for SEPA Credit transaction.

Following SEPA Credit request:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <request-id>30ba2c27-d8bb-47d2-ac02-f1ccf94e162c</request-id>
    <transaction-type>credit</transaction-type>
    <requested-amount currency="EUR">1.01</requested-amount>
        <account-holder>
            <first-name>John</first-name>
            <last-name>Doe</last-name>
            <!-- optional
            <email>john.doe@test.com</email>
            <address>
                <street1>123 anystreet</street1>

```

```

        <city>Brantford</city>
        <country>CA</country>
    </address> -->
</account-holder>
<!-- optional
<order-number></order-number> -->
<!-- optional
<descriptor>test</descriptor> -->
<payment-methods>
    <payment-method name="sepacredit"/>
</payment-methods>
<!-- optional element consumer -->
<consumer>
    <first-name></first-name>
    <last-name></last-name>
</consumer>
<periodic><periodic-type>recurring</periodic-type><sequence-type>recurring</sequence-type></periodic>
    <parent-transaction-id>96468896-3ffb-11e5-919b-005056a64448</parent-transaction-id>
</payment>

```

Following SEPA Credit response:

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>36c1b68b-7a95-4f1b-bcdb-6b84bfbc5b41</merchant-account-id>
    <transaction-id>b03c1d42-3ffb-11e5-919b-005056a64448</transaction-id>
    <request-id>30ba2c27-d8bb-47d2-ac02-f1ccf94e162c</request-id>
    <transaction-type>credit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-08-11T07:36:33.000Z</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully created." severity="information"/>
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
    <parent-transaction-id>96468896-3ffb-11e5-919b-005056a64448</parent-transaction-id>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
    </account-holder>
    <order-number>11153</order-number>
    <descriptor>customerStatement 18009998888</descriptor>
    <payment-methods>
        <payment-method name="sepacredit"/>
    </payment-methods>
    <bank-account>
        <iban>NL53INGB0654422370</iban>
        <bic>INGBNL2A</bic>
    </bank-account>
    <mandate>
        <mandate-id>manid</mandate-id>
        <signed-date>2015-02-02</signed-date>
    </mandate>
    <creditor-id>DE98ZZZ09999999999</creditor-id>
    <periodic>
        <periodic-type>recurring</periodic-type>
        <sequence-type>recurring</sequence-type>
    </periodic>

```

```
<consumer>
<first-name/>
<last-name/>
</consumer>
<provider-transaction-reference-id>3E6B70E7B4</provider-transaction-reference-id>
<instrument-country>NL</instrument-country>
</payment>
```

10.2.9 Cross-Merchant Referencing

It is possible to enable cross payment account references for a merchant in Elastic Engine. It means a transaction from the merchant account A can reference another one from the merchant account B if its processing user has access rights also to the merchant account A. This is a typical scenario for marketplace portals where a consumer can allow to reuse his payment instrument for the further orders without the new authentication process (so-called one-click-payment).

10.2.9.1 Merchant Setup

The general rule is: the processing user used in the referencing transaction must have access rights to the merchant account of the referenced transaction. The access rights to a particular merchant account for the given processing user must be established on the ACL level. In order to enable the cross reference feature for a merchant account the following global merchant parameter must be set to true: **global.crossreference.enabled**.

In order to see the cross referenced transactions in WEP the WEP user must also be granted to access the data of the referenced merchant account. If it's not the case, the cross referenced transactions are not shown inside the payment details list. i.e. the merchants still cannot see the transaction data of other merchants even if their transactions are referencing the transactions of the others. This is a default setup situation for a merchant WEP user.

The billing logic for the cross referenced transactions is unchanged. The cross referenced transactions are billed for the corresponding merchant account used in payment request.

The merchant reconciliation report is unchanged as well. The cross referenced transactions are reported inside the reconciliation file for the corresponding merchant account.

10.2.9.2 XML Samples

Note that “periodic” attribute must be selected to mark which transaction is the first and will be referenced later.

Request for “iDeal” cross-merchant payment

```
POST http://127.0.0.1:8080/engine/rest/paymentmethods/?payment-method-
name=ideal&bic=INGBNL2A HTTP/1.1
Content-Type: application/xml; charset=UTF-8
Content-Length: 1000
Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.3.1 (java 1.5)
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=


<payment xmlns="http://www.elastic-payments.com/schema/payment">
<merchant-account-id>c7f1ba89-73ce-4237-9873-64295913ce7d</merchant-account-id>
<request-id>2f3eb6ce-509c-4eb5-9f3f-527198e9ff75</request-id>
<transaction-type>get-url</transaction-type>
<payment-methods>
<payment-method name="ideal"/>
</payment-methods>
```

```

<requested-amount currency="EUR">2.31</requested-amount>
<order-description>test order</order-description>
<descriptor>customerStatement 18009998888</descriptor>
<bank-account>
  <bic>INGBNL2A</bic>
</bank-account>
<account-holder>
  <first-name>John</first-name>
  <last-name>Doe</last-name>
</account-holder>
<order-number>19317</order-number>
<mandate>
  <mandate-id>manid</mandate-id>
  <signed-date>02-02-2015</signed-date>
</mandate>
<creditor-id>DE98ZZZ09999999999</creditor-id>
<periodic>
  <periodic-type>installment</periodic-type>
  <sequence-type>first</sequence-type>
</periodic>
</payment>

```

Response for “iDeal” cross-merchant payment – debit on iDeal site

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
  <merchant-account-id>c7f1ba89-73ce-4237-9873-64295913ce7d</merchant-account-id>
  <transaction-id>713e3527-1a4b-11e5-b3a3-0050b65c678c</transaction-id>
  <request-id>2f3eb6ce-509c-4eb5-9f3f-527198e9ff75</request-id>
  <transaction-type>get-url</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2015-06-24T10:31:58.000+02:00</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <requested-amount currency="EUR">2.31</requested-amount>
  <account-holder>
    <first-name>John</first-name>
    <last-name>Doe</last-name>
  </account-holder>
  <order-number>19317</order-number>
  <descriptor>customerStatement 18009998888</descriptor>
  <payment-methods>
    <payment-method url="https://idealtest.secure-ing.com/ideal/issuerSim.do?trxd=0050000101476386&ideal=prob" name="ideal"/>
  </payment-methods>
  <bank-account>
    <bic>INGBNL2A</bic>
  </bank-account>
  <mandate>
    <mandate-id>manid</mandate-id>
    <signed-date>02-02-2015</signed-date>
  </mandate>
  <creditor-id>DE98ZZZ09999999999</creditor-id>
  <periodic>
    <periodic-type>installment</periodic-type>
    <sequence-type>first</sequence-type>
  </periodic>
</payment>

```

```
</periodic>
</payment>
```

Merchant notification after debit

```
<?xml version="1.0" encoding="UTF-8"?><payment xmlns="http://www.elastic-payments.com/schema/payment"><merchant-account-id>c7f1ba89-73ce-4237-9873-64295913ce7d</merchant-account-id><transaction-id>739faf84-1a4b-11e5-b3a3-0050b65c678c</transaction-id><request-id>2f3eb6ce-509c-4eb5-9f3f-527198e9ff75</request-id><transaction-type>debit</transaction-type><transaction-state>success</transaction-state><completion-time-stamp>2015-06-24T10:32:00.000+02:00</completion-time-stamp><statuses><status code="201.1126" description="ideal:Successful confirmation received from the bank." severity="information"/></statuses><requested-amount currency="EUR">2.31</requested-amount><account-holder><first-name>Hr</first-name><last-name>E G H Küppers en/of MW M.J. Küpp</last-name></account-holder><order-number>19317</order-number><descriptor>customerStatement 18009998888</descriptor><payment-methods><payment-method name="ideal"/></payment-methods><mandate><mandate-id>manid</mandate-id><signed-date>0007-08-08T00:00:00.000+01:00</signed-date></mandate><creditor-id>DE98ZZZ09999999999</creditor-id><api-id>---</api-id><periodic><periodic-type>installment</periodic-type><sequence-type>first</sequence-type></periodic><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>bZHaJgxjFpyO7VnpTwaTwA3ynM8=</DigestValue></Reference></SignedInfo><SignatureValue>ZWOejEbW1k4xfW85qTWSfaXFEPL4i1QuopVjmRctiZ844DYqninTi7q06xRM4EbC3hBi5da2/yn1 EGB06T3Z8FrImQtOgoIV5iGBqWqDZMF+EB7UTwMXU+VQMCzOLFBLgDacVCMHCuEvcWVKJ05y+SD0abg8rOX1P/oaaLmHYSSkF7TC/s0BcGI7pb3VSW3KhOAIFx2t0FEEioE2+t7GUgVNPs8FlxnWHZ2yQIfbIEEMmFD5UrAmHJ/uUj8c4qmGgQ2EcdSSvUO8gy1LZisstQrN8GhwyvaimrSv/QKwvAAwTnO+ms/TVEGoKuaKCZzxHPGquQFQ8pG5cdG+m/JjQ==</SignatureValue><KeyInfo><X509Data><X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBACtRvcm9udG8xIjAgBgNVBAoTGVdpPCMvJYXjkIEVsYXN0aWMgUGF5bWVu dHMxExARBgNVBAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNj0MDQ0NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBACtB1Rvcm9udG8xIjAgBgNVBAoTGVdpPCMvJYXjkIEVsYXN0aWMgUGF5bWVu dHMxExARBgNVBAsTCk9wZXJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwHhcNMTEwNj0MDQ0NDA5WhcNMTQwMzIwMDQ0NDA5WjB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBACtZJhdGlvbnnMxEzARBgNVBAMTCk1hbm9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK AoIBAQCc8rTt4N5fNeVzlsRgOKXDE2YUSfJx7xBozFZ3Vh3XQyy3IpluEfZz7004k4HeonTxCN etBvJ9rgNc0Cxrk/euMj3pOurE9WYN2eAXC0r5pUIAZhIAhSxUsaIf3JKBxf7gDAik5d8RT5HaJV 4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3lyLwhDbYlyU6j4fMKyHlAMGzW7VgKD 2pqu6BRysqUvdEEAvW2OmyVqGvypkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvnrWbIzyQuyUyDeUb QmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14rSXRMGzeJ+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEB ADgkuN/e2IFy7JXdbjNjbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZr mk/KqOyUL+unVcR+APjxX4KP25sdplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m 02qvGoK8UA/IRblQ6DZ9hEKV5VQKIMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KChpfw2j8G+dDgU jlp9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRK0/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3 x/QspKOjYn1Kzw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate></X509Data></KeyInfo></Signature></payment>
```

Now this is a request for SEPA Credit where periodic type is “recurring” which means that this is a referencing payment and note that the “parent-transaction-id” is same as the one from the “first” (parent) payment in response.

Request for SEPA Credit cross-merchant payment

```

POST http://127.0.0.1:8080/engine/rest/paymentmethods/?payment-method-
name=sepacredit&bic=WIREDEMMXXX&periodic-type=recurring&sequence-type=recurring HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Content-Length: 1033
Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.3.1 (java 1.5)
Authorization: Basic dGVhbWNpdHk6dGVhbWNpdHk=


<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>e7c87536-2e7d-4fcf-abc8-9bb1d81907ca</merchant-account-id>
    <request-id>3864ca4a-eaac-4a65-968c-047676fd6101</request-id>
    <transaction-type>credit</transaction-type>
    <requested-amount currency="EUR">1.01</requested-amount>
        <account-holder>
            <first-name>John</first-name>
            <last-name>Doe</last-name>
        </account-holder>
    <payment-methods>
        <payment-method name="sepacredit"/>
    </payment-methods>
    <mandate>
        <mandate-id>mandid</mandate-id>
        <signed-date>2015-01-05</signed-date>
    </mandate>
    <creditor-id>XY98ZZZ09999999999</creditor-id>
    <consumer>
        <first-name></first-name>
        <last-name></last-name>
    </consumer>
    <periodic><periodic-type>recurring</periodic-type>
        <sequence-type>recurring</sequence-type></periodic>
        <parent-transaction-id>3aa87578-1a4c-11e5-b3a3-0050b65c678c</parent-transaction-id>
    </parent-transaction-id>
</payment>

```

Response for SEPA Credit cross-merchant payment

```

<payment xmlns="http://www.elastic-payments.com/schema/payment">
    <merchant-account-id>e7c87536-2e7d-4fcf-abc8-9bb1d81907ca</merchant-account-id>
    <transaction-id>3da64fb6-1a4c-11e5-b3a3-0050b65c678c</transaction-id>
    <request-id>3864ca4a-eaac-4a65-968c-047676fd6101</request-id>
    <transaction-type>credit</transaction-type>
    <transaction-state>success</transaction-state>
    <completion-time-stamp>2015-06-24T10:37:39.000+02:00</completion-time-stamp>
    <statuses>
        <status code="201.0000" description="The resource was successfully created." severity="information"/>
    </statuses>
    <requested-amount currency="EUR">1.01</requested-amount>
    <parent-transaction-id>3aa87578-1a4c-11e5-b3a3-0050b65c678c</parent-transaction-id>
    <account-holder>
        <first-name>John</first-name>
        <last-name>Doe</last-name>
    </account-holder>

```

```

<order-number>28862</order-number>
<descriptor>customerStatement 18009998888</descriptor>
<payment-methods>
  <payment-method name="sepacredit"/>
</payment-methods>
<bank-account>
  <iban>NL53INGB0654422370</iban>
  <bic>INGBNL2A</bic>
</bank-account>
<mandate>
  <mandate-id>mandid</mandate-id>
  <signed-date>2015-01-05</signed-date>
</mandate>
<creditor-id>XY98ZZZ09999999999</creditor-id>
<periodic>
  <periodic-type>recurring</periodic-type>
  <sequence-type>recurring</sequence-type>
</periodic>
<consumer>
  <first-name/>
  <last-name/>
</consumer>
<provider-transaction-reference-id>7BE65FA70E</provider-transaction-reference-id>
<instrument-country>NL</instrument-country>
</payment>

```

SEPA Credit notification for merchant

```

<?xml version="1.0" encoding="UTF-8"?><payment xmlns="http://www.elastic-payments.com/schema/payment"><merchant-account-id>e7c87536-2e7d-4fcf-abc8-9bb1d81907ca</merchant-account-id><transaction-id>3ddf60d2-1a4c-11e5-b3a3-0050b65c678c</transaction-id><request-id>3864ca4a-eaac-4a65-968c-047676fd6101</request-id><transaction-type>credit</transaction-type><transaction-state>success</transaction-state><completion-time-stamp>2015-06-24T10:37:40.000+02:00</completion-time-stamp><statuses><status code="201.0000" description="bank:The resource was successfully created." provider-transaction-id="61ab74011ab6e13a797b00e47bf01106" severity="information"/></statuses><requested-amount currency="EUR">1.010000</requested-amount><account-holder><first-name>John</first-name><last-name>Doe</last-name></account-holder><order-number>28862</order-number><descriptor>customerStatement 18009998888</descriptor><payment-methods><payment-method name="sepacredit"/></payment-methods><mandate><mandate-id>mandid</mandate-id><signed-date>2015-01-05T00:00:00.000+01:00</signed-date></mandate><creditor-id>XY98ZZZ09999999999</creditor-id><api-id>---</api-id><periodic><periodic-type>recurring</periodic-type><sequence-type>recurring</sequence-type></periodic><provider-transaction-reference-id>7BE65FA70E</provider-transaction-reference-id><instrument-country>NL</instrument-country><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>aufM08TY4Xqa70knFRr93HXsy7l=</DigestValue></Reference></SignedInfo><SignatureValue>cFkjLvNakh/1lg+0kBhgSYvWymv8+09iPHeidjXy/J2Z8xptwXj6429cC/qYJxqX7gRCXhVMPBRvkw4dXOn+PsXGHkPcqDR2O1UshtgxSMoqaiOU+aHrZAS0UORW/23iSXl4ww9yYknphDQknCCXAkEl1suX6eX34tUi1WNf1IVchDm6Og5wi5m1y7YaV5H5VT+YFsbKf5/wn+RMpDol97R6gRk97gKA5rdjiU4atMRtyGz3aOkv2Gxejnh8dvB6PXcieK2MrsocOjEyoOvN2ZN5dzhgbOlvBrwo+Jc8E74tWti

```

22d5HUDxH1SwTXVnOJHylwxZP7Nog1LfUvpzzg==</SignatureValue><KeyInfo><X509Data><X509SubjectName>CN=Manoj Sahu,OU=Operations,O=Wirecard Elastic Payments,L=Toronto,ST=ON,C=CA</X509SubjectName><X509Certificate>MIIDcDCCAligAwIBAgIETgQWGTANBgkqhkiG9w0BAQUFADB6MQswCQYDVQQGEwJDQTELMAkGA1UECBMCT04xEDAOBgNVBACTB1Rvcn9udG8xIjAgBgNVBAoTGVdpcmVjYXJkIEVsYXN0aWMgUGF5bVVudHMxEzARBgNVBAAsTCk9wZXJhdGlvbnMxEzARBgNVBAMTck1hbh9qIFNhaHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCc8rTt4N5fNeVzlsRgOXKDE2YUSfJx7xBozFZ3Vh3XQyy3lpluEfZz7004k4HeonfTxCNetBvJ9rgNc0Cxrk/euMj3pOUrE9WYN2eAXC0r5pUIAZhIANsUSAf3JKBxf7gDAik5d8RT5HaJV4n5cXJQ/uhAEYU3EGN/74UrD2UsOYD3VBXTJS5VgSi/c3lyLwhDbYIyU6j4fMKyHIIAMGzW7VgKD2pqu6BRysqUVdEEAvW2OmyVqGVyPkm87EiHSMMSSar3CvYYxYqBN2KBUjabkvNRWblzyQuyUyDeUbQmhVQKL0WIMb5ev65m2VjGyDTGL5jfB14fSXRMGzej+LAgMBAAEwDQYJKoZIhvcNAQEFBQADggEBADgkuN/e2Ify7JXdbjNjbKBd3HLvFvK87dv8qQ+HK4qfCxYXh6aYhbKHJSA6C2pbOD3HBXoyovZrmk/KqOyUL+unVcR+APjxX4KP25sdkplgmeQ47CWxtKAHZUTtWwAVI/WhsX89SSucBfIS5TJ54e7m02qvGoK8UA/IRbIQ6DZ9hEKV5VQKiMx3ubwwHGXFOWz2fKmeZBuTeY+HiTEH8KCHpfw2j8G+dDgUjl9LvjVNmJzfNBBk1Si0d/rhXmMzVSKj08tp1sPRKO/sJtJZBzQajpnsZ9NFfoJNdG13AzYwDP3x/QspK0jYn1KZw1qz524VWoQoueR8Xj30A2jntA=</X509Certificate></X509Data></KeyInfo></Signature></payment>

11 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 facilitates 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.
paysafecard	Paysafecard	Electronic payment method for predominantly online shopping and is based on a pre-pay system.
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	Guaranteede 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
trustpay	Trust Pay	An alternative payment method aggregating banks in CEE region.
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.

12 Appendix B: Transaction Types

transaction-type	name	description
adjust-capture	Adjust Capture	Adjusts funds from the [Card Holder] to [Merchant Account].
adjust-offline-purchase	Adjust Offline Purchase	Adjusts funds from the [Card Holder] to [Merchant Account].
adjust-purchase	Adjust Purchase	Adjusts funds from the [Card Holder] to [Merchant Account].
adjustment-cr	Adjustment Credit	Adjusts (credits) funds to the [Merchant Account].
adjustment-dr	Adjustment Debit	Adjusts (debits) funds from the [Merchant Account].
authorization	Authorization	Reserves funds from the [Account Holder]'s account.
authorization-only	Authorization Only	Verifies the card's validity without leaving an authorized amount.
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.
batch-upload	Batch Upload	Acquirer file update advice.
capture-authorization	Capture Authorization	Takes funds from the [Card Holder]'s account. Must follow an [Authorization] or [Authorization Supplementary] chain.
capture-final-auth	Capture Final Authorization	Takes funds from the [Card Holder]'s account. Must follow an [Final-Authorization] chain.
capture-preauthorization	Capture Preauthorization	Takes funds from the [Card Holder]'s account. Must follow an [Preauthorization] chain.
check-enrollment	3D Check Enrollment	Verifies the 3D participation of a card using card scheme directory server and/or ACS.
check-payer-response	3D Check Payer Response	Verifies that the cardholder authenticated the transaction within the 3-D Secure Program.
check-risk	Check Risk	Checks the risk profile of the transaction information, without submitting a payment.
check-signature	Check Signature	Validates digital signature for a [Payment Page] transaction.
credit	Credit	Moves funds from the [Merchant] to the [Account Holder].
credit-request	Credit Request	A request to move funds from the [Merchant] to the [Account Holder]. For this [Payment Method], the actual action is taken with the subsequent "credit".
credit-return	Credit Return	An adjustment made to a [Credit] as a result of a dispute. Moves funds from the [Account Holder] to the [Merchant].
debit	Debit	Moves funds from the [Account Holder] to the [Merchant].

debit-return	Debit Return	An adjustment made to a [Debit] as a result of a dispute. Moves funds from the [Merchant] to the [Account Holder].
debit-return-succ-rep	Debit Return Rep Succ	A successful representation made on a Debit Return.
debit-return-unsucc-rep	Debit Return Rep Unsucc	An unsuccessful representation made on a Debit Return.
deposit	Deposit	Moves funds from the [Account Holder] to the [Merchant].
detokenize	Detokenize	Provides sensitive Credit Card Data from a unique [Card Token].
final-authorization	Final Authorization	Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture-final-authorization] on this transaction.
first-cb	First Chargeback	First Chargeback
first-cb-rep	First Chargeback Rep	First Chargeback Representment
first-cb-succ-rep	First Chargeback Rep Successful	First Chargeback Representment Success
first-cb-unsucc-rep	First Chargeback Rep Unsuccessful	First Chargeback Representment Unsuccessful
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.
offline-purchase	Offline Purchase	Takes funds from the [Card Holder]'s account. A successful Purchase transaction completed offline at Point Of Sale.
original-credit	Original Credit	Gives funds to the [Cardholder]'s account, NOT referring to an eligible [Purchase] or [Capture].
pending-credit	Pending Credit	A request to move funds from the [Merchant] to the [Account Holder]. For this [Payment Method], the actual action is taken with subsequent transaction type [credit].
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.
preauthorization	Preauthorization	Reserves funds from the [Card Holder]'s account.
precheckout	Precheckout	Requests for precheckout wallet information of [Account Holder] from [Wallet Provider]
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, referring to an eligible [Capture].
refund-debit	Refund Debit	Moves funds from the [Merchant] to the [Account Holder]. For this [Payment Method], typically follows a 'refund-request'.
refund-off-purchase	Refund Offline Purchase	Gives funds to the [Card Holder]'s account, referring to an eligible [Offline Purcahse].
refund-purchase	Refund Purchase	Gives funds to the [Card Holder]'s account, referring to an eligible [Purchase].
refund-request	Refund request	A request to give funds to the [Card Holder]'s account, referring to an eligible [Capture/Purchase].
request-checkout	Request for Checkout	Requests for checkout from [Wallet Provider]
retrieval-request	Retrieval Request	Retrieval Request
retrieval-request-rep	Retrieval Request Rep	Retrieval Request Representment

reversal-authorization	Reversal Authorization	Frees reserved funds from the [Card Holder]'s account.
reversal-purchase	Reversal Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-refund-capture	Reversal Refund Capture	Frees reserved funds from the [Card Holder]'s account due to an [Refund Capture].
reversal-refund-off-purchase	Reversal Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Offline Purchase].
reversal-refund-purchase	Reversal Refund Purchase	Frees reserved funds from the [Card Holder]'s account due to a [Refund Purchase].
reversal-standalone-refund	Reversal Standalone Refund	Frees reserved funds from the [Card Holder]'s account due to a [Standalone Refund].
reversal-void-auth	Reversal Void Authorization	Frees reserved funds from the [Card Holder]'s account.
reversal-void-capture	Reversal Void Capture	Frees reserved funds from the [Card Holder]'s account.
reversal-void-off-purchase	Reversal Void Offline Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-purchase	Reversal Void Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-capture	Reversal Void Refund Capture	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-off-purch	Reversal Void Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-purchase	Reversal Void Refund Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-st-refund	Reversal Void Standalone Refund	Frees reserved funds from the [Card Holder]'s account.
second-cb	Second Chargeback	Second Chargeback
second-cb-rep	Second Chargeback Rep	Second Chargeback Representment
second-cb-sucess-rep	Second Chargeback Rep Successful	Second Chargeback Representment Success
second-cb-unsucc-rep	Second Chargeback Rep Unsuccessful	Second Chargeback Representment Unsuccessful
settlement	Settlement	Acquirer reconciliation request.
settlement-close	Settlement Close	Settlement Close - It's sent after the upload
standalone-refund	Standalone Refund	Gives funds to the [Card Holder]'s account without referring to an eligible [Capture] or [Purchase]
tc-upload	Tc Upload	After a transaction is completed, the terminal may update some ICC data elements
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-auth-supplementary	Void Authorization Supplementary	Voids an upwardly adjustment of an existing authorization.
void-authorization	Void Authorization	Frees reserved funds from the [Card Holder]'s account due to an [Authorization] or chain of [Authorization Supplementary]s.
void-capture	Void Capture	Frees reserved funds from the [Card Holder]'s account due to a [Capture].
void-capture-final-auth	Void Capture Final Authorization	Frees reserved funds from the [Card Holder]'s account due to a [Capture-final-authorization].
void-capture-preauthorization	Void Capture Preauthorization	Frees reserved funds from the [Card Holder]'s account due to a [Capture-preauthorization].
void-credit	Void Credit	Frees reserved funds from the [Card Holder]'s account due to a [Credit].
void-debit	Void Debit	Frees reserved funds from the [Card Holder]'s account due to a [Debit].

void-final-authorization	Void Final Authorization	Frees reserved funds from the [Card Holder]'s account due to an [Final-authorization].
void-offline-purchase	Void Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to a [Offline Purchase].
void-original-credit	Void Original Credit	Frees reserved funds from the [Card Holder]'s account due to a [Orginal Credit].
void-pending-credit	Void Pending Credit	A request to cancel pending-credit transactions. For this [Payment Method], the actual action is taken with transaction type [void-pending-credit].
void-pending-debit	Void Pending Debit	A request to cancel pending-debit transactions. For this [Payment Method], the actual action is taken with transaction type [void-pending-debit].
void-preauthorization	Void Preauthorization	Frees reserved funds from the [Card Holder]'s account due to an [Preauthorization].
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-debit	Refund Debit Adjustment	An adjustment made to a successful [Refund Debit] after the result was reconciled with the bank. Moves funds from the [Account Holder] to the [Merchant].
void-refund-off-purchase	Void Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Offline Purchase].
void-refund-purchase	Void Refund Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Purchase].
void-standalone-refund	Void Standalone Refund	Frees reserved funds from the [Card Holder]'s account due to a [Standalone Refund].

Credit Card transaction types

transaction-type	name	description
adjust-capture	Adjust Capture	Adjusts funds from the [Card Holder] to [Merchant Account].
adjust-offline-purchase	Adjust Offline Purchase	Adjusts funds from the [Card Holder] to [Merchant Account].
adjust-purchase	Adjust Purchase	Adjusts funds from the [Card Holder] to [Merchant Account].
authorization	Authorization	Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture] on this transaction.
authorization-only	Authorization Only	Verifies the card's validity without leaving an authorized amount.
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.
batch-upload	Batch Upload	Acquirer file update advice.
capture-authorization	Capture Authorization	Takes funds from the [Card Holder]'s account. Must follow an [Authorization] or [Authorization Supplementary] chain.
capture-final-auth	Capture Final Authorization	Takes funds from the [Card Holder]'s account. Must follow an [Final-Authorization] chain.
capture-preauthorization	Capture Preauthorization	Takes funds from the [Card Holder]'s account. Must follow an [Preauthorization] chain.
check-enrollment	3D Check Enrollment	Verifies the 3D participation of a card using card scheme directory server and/or ACS.
check-payer-response	3D Check Payer Response	Verifies that the cardholder authenticated the transaction within the 3-D Secure Program.

check-risk	Check Risk	Checks the risk profile of the transaction information, without submitting a payment.
check-signature	Check Signature	Validates digital signature for a [Payment Page] transaction.
credit	Credit	Moves funds from the [Merchant] to the [Account Holder].
detokenize	Detokenize	Provides sensitive Credit Card Data from a unique [Card Token].
final-authorization	Final Authorization	Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture-final-authorization] on this transaction.
first-cb	First Chargeback	First Chargeback
first-cb-rep	First Chargeback Rep	First Chargeback Representment
first-cb-succ-rep	First Chargeback Rep Successful	First Chargeback Representment Success
first-cb-unsucc-rep	First Chargeback Rep Unsuccessful	First Chargeback Representment Unsuccessful
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.
offline-purchase	Offline Purchase	Takes funds from the [Card Holder]'s account. A successful Purchase transaction completed offline at Point Of Sale.
original-credit	Original Credit	Gives funds to the [Cardholder]'s account, NOT referring to an eligible [Purchase] or [Capture].
preauthorization	Preauthorization	Reserves funds from the [Card Holder]'s account. 7 days left to conduct a [Capture-preauthorization] on this transaction.
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, referring to an eligible [Capture].
refund-off-purchase	Refund Offline Purchase	Gives funds to the [Card Holder]'s account, referring to an eligible [Offline Purcahse].
refund-purchase	Refund Purchase	Gives funds to the [Card Holder]'s account, referring to an eligible [Purchase].
refund-request	Refund request	A request to give funds to the [Card Holder]'s account, referring to an eligible [Capture/Purchase].
retrieval-request	Retrieval Request	Retrieval Request
retrieval-request-rep	Retrieval Request Rep	Retrieval Request Representment
reversal-authorization	Reversal Authorization	Frees reserved funds from the [Card Holder]'s account.
reversal-purchase	Reversal Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-refund-capture	Reversal Refund Capture	Frees reserved funds from the [Card Holder]'s account due to an [Refund Capture].
reversal-refund-off-purchase	Reversal Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Offline Purchase].
reversal-refund-purchase	Reversal Refund Purchase	Frees reserved funds from the [Card Holder]'s account due to a [Refund Purchase].
reversal-standalone-refund	Reversal Standalone Refund	Frees reserved funds from the [Card Holder]'s account due to a [Standalone Refund].
reversal-void-auth	Reversal Void Authorization	Frees reserved funds from the [Card Holder]'s account.
reversal-void-capture	Reversal Void Capture	Frees reserved funds from the [Card Holder]'s account.
reversal-void-off-purchase	Reversal Void Offline Purchase	Frees reserved funds from the [Card Holder]'s account.

reversal-void-purchase	Reversal Void Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-capture	Reversal Void Refund Capture	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-off-purch	Reversal Void Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-refund-purchase	Reversal Void Refund Purchase	Frees reserved funds from the [Card Holder]'s account.
reversal-void-st-refund	Reversal Void Standalone Refund	Frees reserved funds from the [Card Holder]'s account.
second-cb	Second Chargeback	Second Chargeback
second-cb-rep	Second Chargeback Rep	Second Chargeback Representment
second-cb-succ-rep	Second Chargeback Rep Successful	Second Chargeback Representment Success
second-cb-unsucc-rep	Second Chargeback Rep Unsuccessful	Second Chargeback Representment Unsuccessful
settlement	Settlement	Acquirer reconciliation request.
settlement-close	Settlement Close	Settlement Close - It's sent after the upload
standalone-refund	Standalone Refund	Gives funds to the [Card Holder]'s account without referring to an eligible [Capture] or [Purchase]
tc-upload	Tc Upload	After a transaction is completed, the terminal may update some ICC data elements
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-auth-supplementary	Void Authorization Supplementary	Voids an upwardly adjustment of an existing authorization.
void-authorization	Void Authorization	Frees reserved funds from the [Card Holder]'s account due to an [Authorization] or chain of [Authorization Supplementary]s.
void-capture	Void Capture	Frees reserved funds from the [Card Holder]'s account due to a [Capture].
void-capture-final-auth	Void Capture Final Authorization	Frees reserved funds from the [Card Holder]'s account due to a [Capture-final-authorization].
void-capture-preauthorization	Void Capture Preauthorization	Frees reserved funds from the [Card Holder]'s account due to a [Capture-preauthorization].
void-credit	Void Credit	Frees reserved funds from the [Card Holder]'s account due to a [Credit].
void-final-authorization	Void Final Authorization	Frees reserved funds from the [Card Holder]'s account due to an [Final-authorization].
void-offline-purchase	Void Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to a [Offline Purchase].
void-original-credit	Void Original Credit	Frees reserved funds from the [Card Holder]'s account due to a [Orginal Credit].
void-preauthorization	Void Preauthorization	Frees reserved funds from the [Card Holder]'s account due to an [Preauthorization].
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-off-purchase	Void Refund Offline Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Offline Purchase].
void-refund-purchase	Void Refund Purchase	Frees reserved funds from the [Card Holder]'s account due to an [Refund Purchase].
void-standalone-refund	Void Standalone Refund	Frees reserved funds from the [Card Holder]'s account due to a [Standalone Refund].

Transaction Types matrix

	preauthoriza tion	authorizat ion	authorizati on-only	capture- authorizat ion	check- signatu re	cred it	deb it	ge t- url	pendi ng- debit	pre- checko ut	pendi ng- credit	purcha se	refun d- captu re	refun d- debit	void- authorizat ion	void- pendi ng- debit	
Alipay Cross- border					X		X	X	X						X		
Alipay Domestic						X		X	X	X					X		
Carrier Billing						X		X	X						X		
cimbclicks					X		X	X									
Chinapay					X		X	X							X		
direktüberwei sung						X		X	X								
eplatby						X		X	X								
epay.bg								X	X								
eps						X	X	X	X								
ideal						X		X	X								
masterpass		X	X			X		X	X		X		X				
maybank2u						X		X	X								
moneta_ru						X	X	X	X	X							
PayPal	X	X	X	X	X	X	X	X	X			X		X	X	X	
Paysafecard		X			X	X		X								X	
ratepay-elv	X				X	X	X		X						X		X
ratepay- invoice		X			X	X	X		X						X		X
SEPA		X					X	X		X		X					X
sofortbanking						X		X	X								
trustpay						X		X	X								
yandex						X	X	X	X						X		
Skrill Wallet						X	X	X	X						X		

13 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.

14 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

15 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	VOID is no longer possible. Transaction is currently being processed and / or has already been sent to the bank.
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 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 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	Malformatted timeout string	Malformatted 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.
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 with Tokenization(remote) Server	Failed to synchronize with Tokenization(remote) 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 maximum required for this Merchant Account.
Error	400.1119	Velocity: Velocity Limit Reached, Merchant Account	The transaction was refused because this 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.
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 card number	Mismatch card type and 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.
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 merchant account identifiers supplied	Provide one of merchant account id, merchant 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

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 applies)	The debtor IBAN is not allowed based on country restriction
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 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 debtor to his bank	No authorisation from 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
Error	500.3002	Risk: General risk management rejection	Risk: General risk management rejection
Error	500.3004	PaySafeCard timeout.	Cards allocation will be cancelled.
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

16 Appendix F: Countries and Currencies

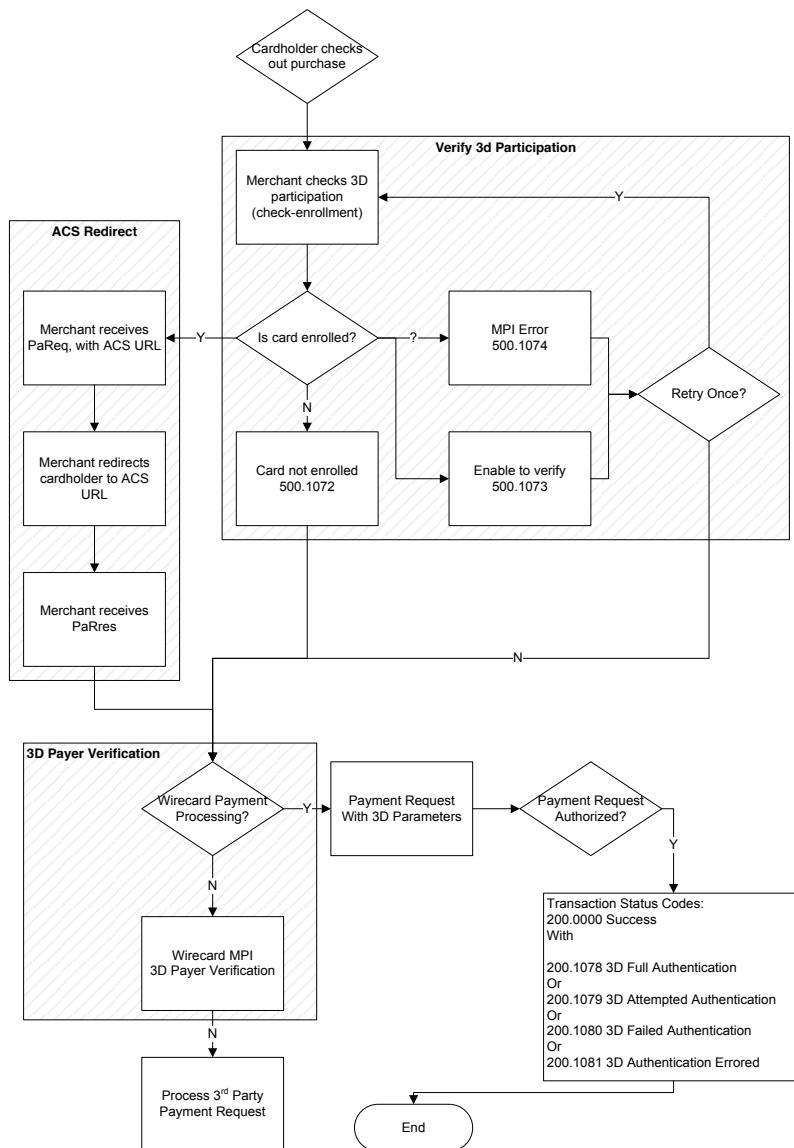
Countries follow the ISO 3166-1-alpha-2 code.

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

17 Appendix G: 3-D-Secure Process Flow



18 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 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 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.
PATRes	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 an 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 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.

19 Appendix I: iDEAL Participating Banks

BIC	Name
ABNANL2A	ABN Amro Bank
ASBNL21	ASN Bank
FRBKNL2L	Friesland Bank
INGBNL2A	ING
RABONL2U	Rabobank
SNSBNL2A	SNS Bank
RGGINL21	Regio Bank
TRIONL2U	Triodos Bank
FVLBNL22	Van Lanschot Bankiers
KNABNL2H	Knab

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

EE	Status	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 employed 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.

21 Appendix K: Test Access Data and Credentials

Merchant Account ID	Merchant Account	Payment Method	EE Https User	EE Https password	Use URL for XML services	Secret key Hosted Payment Page (HPP)	HPP URL
1b3be510-a992-48aa-8af9-6ba4c368a0ac	Wirecard CC/EFT DemoCC no CVC	Credit Card - Non-3D (Demo)	70000-APIDEM O-CARD	ohysS0-dvfMx	https://api-test.wirecard.com/engine/rest/payments/	22070686-8ca2-4a97-a55e-6853bfae470a	https://api-test.wirecard.com/engine/hpp/
9105bb4f-ae68-4768-9c3b-3eda968f57ea	Wirecard CC/EFT TestCC no CVC	Credit Card - Non-3D (Test)	70000-APILUHN -CARD	8mhwav KVb91T		d1efed51-4cb9-46a5-ba7b-0fdc87a66544	
33f6d473-3036-4ca5-acb5-8c64dac862d1	Wirecard CC/EFT Simu3D no CVC	Credit Card - 3D (Test)				9e0130f6-2e1e-4185-b0d5-dc69079c75cc	
4c901196-eff7-411e-82a3-5ef6b6860d64	WD SEPA Test	SEPA	70000-APITEST -AP	qD2wzQ_hrcI8	https://api-test.wirecard.com/engine/rest/paymentmethods/	ecdf5990-0372-47cd-a55d-037dccfe9d25	

f7bb4ccc-1acb-48d2-99e6-ba962bb0b74d	WD Skrill Wallet Test	SKRILL Wallet				e9531d9d-fa88-47f9-aa5d-bcf5f6bbc9e	
adb45327-170a-460b-9810-9008e9772f5f	Ideal direct	Ideal				1b9e63b4-c132-42c3-bcbd-2d2e47ae7154	
f19d17a2-01ae-11e2-9085-005056a96a54	Sofortüberweisung	SoÜ				ad39d9d9-2712-4abd-9016-cdeb60dc3c8f	
1eb31434-2b0f-48dc-86f6-7926a67e7f25	moneta_ru	moneta_ru				3361ef1c-0c17-48e4-9e2c-18d4c3346944	
9abf05c1-c266-46ae-8eac-7f87ca97af28	PayPal	PayPal				5fca2a83-89ca-4f9e-8cf7-4ca74a02773f	
c35733ea-ca79-4781-a5c3-74ce9746eac9	RatePAY Test Merchant	RatePay				e27da925-69cf-4e9e-a716-c9a001600199	
8bc8ed6d-81a8-43be-bd7b-75b008f89fa6	WireCard MasterPass	MasterPass				87e1c187-363f-41d8-abab-7bae0fb03eca	
4c0de18e-4c20-40a7-a5d8-5178f0fe95bd	WireCard Paysafecard	Paysafecard				bb1f2975-827b-4aa8-bec6-405191d85fa5	
fe6c560b-5f28-4e0a-9bde-cee067f97ed6	WireCard TrustPay	Trustpay	0000003 16D6E4F E0- EMWDM S	SmqJNf-FQ1Qg			
5f15a3cf-5cea-4dac-873c-2e5e46225cdf	EE - P24	P24	Merchant_Test_User_MUC	TestXAPTER_1			

Test Credit Card 3D

Brand	Card Number	Expiry Date	CVC	Auth. result	Enroll m. Result	ACS Passwo rd	ACS personal message	Response Code	Advice
VISA	4012000300001003	Jan 19	003	Y	Y	wirecard	Testing "Authentication approved"	201.0000	Approved or completed successfully
Mastercard	5413330300001006	Jan 19	006						
Maestro	6799860300001000003	Jan 19	003						

VISA	4012000300002001	Jan 19	001	A	Y		No cardholder interaction	201.0000	Approved or completed successfully
Mastercard	5413330300002004	Jan 19	004						
Maestro	6799860300002000	Jan 19	002						
VISA	4012000300003009	Jan 19	009	N	Y	wirecard	Testing "Authentication failed"	500.1999	Verification failed. Cardholder failed or cancelled 3-D Secure authentication.
Mastercard	5413330300003002	Jan 19	002						
Maestro	6799860300003000	Jan 19	001						
VISA	4012000300004007	Jan 19	007	U	Y	wirecard	Testing "Unable to authenticate"	201.0000	Approved or completed successfully / E-Commerce only
Mastercard	5413330300004000	Jan 19	999						
Maestro	6799860300004000	Jan 19	999						
VISA	4012000300005004	Jan 19	004	E	Y	wirecard	Testing "Authentication failed due to system error"	201.0000	Approved or completed successfully / E-Commerce only
Mastercard	5413330300005007	Jan 19	007						
Maestro	6799860300005000	Jan 19	009						
VISA	4012000300006002	Jan 19	002		N			500.1072	Cardholder not participating. This card is eligible but not enrolled in the 3-D Secure program.
Mastercard	5413330300006005	Jan 19	005						
Maestro	6799860300006000	Jan 19	008						
VISA	4012000300007000	Jan 19	999		U			500.1073	Unable to verify enrollment.
Mastercard	5413330300007003	Jan 19	003						
Maestro	6799860300007000	Jan 19	007						
VISA	4012000300008008	Jan 19	008		E			5.001074	A system error prevented enrollment from completing.
Mastercard	5413330300008001	Jan 19	001						

Maestro	6799860300008000 006	Jan 19	006						
---------	-------------------------	-----------	-----	--	--	--	--	--	--