



B-Gateway

User guide

Borgun hf.
+354 560 1699
www.borgun.com

History

2007-09-18	PF	Added field lengths and data filed description
2009-05-04	BA	RRN search added to getTransactionList
2009-05-04	BA	Added disputed transaction chapter
2009-05-04	BA	Added settlement chapter
2009-08-15	BA	Added Card Information chapter
2009-08-28	BA	Added fraud transaction chapter
2010-05-06	BA	Clarification on minor issues
2011-01-15	BA	Numpayments/FirstPayments removed
2011-02-17	BA	Added getVirtualCard
2011-04-05	BA	RRN recommended value changed
2011-07-19	BA	Added FAQ, new template for look
2011-09-06	BA	Added getSettlementChargeback, Batch/Transaction changes
2011-11-30	BA	Added Partial Reversal
2012-05-08	BA	Added dynamic descriptors
2012-06-12	BA	Added DetailData and Passenger Itinerary
2012-09-12	BA	Added text to dynamic descriptors
2014-06-26	BA	Update
2014-08-25	BA	Added Credit tag to getTransactionList
2014-08-26	BA	Added CVCResult description
2015-01-26	BA	Added SPR to chargeback description
2015-03-20	BA	Added text about zero amount pre-authorization
2016-05-18	BA	Descriptors revisited

Table of Contents

About B-Gateway.....	4
Key features	4
Get authorizations	4
Cancel authorizations.....	4
Create new batches	4
Get batch authorizations	4
Get processing results.....	4
Get transaction lists	4
Send detail data	4
Introduction to batches	5
How does B-Gateway work?	5
Data Storage and Manipulation	6
Data types.....	6
Getting authorizations	7
Normal purchase transaction.....	7
Refund	7
Partial Reversal	7
Pre-authorizations	7
Dynamic Descriptors	8
Getting authorizations – XML explanations	10
Getting authorizations – XML examples	13
Cancelling authorizations.....	14
Cancelling authorizations – XML explanations.....	15
Cancelling authorizations – XML examples	16
Virtual cardnumbers	16
Virtual cardnumbers – XML explanations	16
Cancelling authorizations – XML examples	17
Creating new batches	17
Batch numbers controlled by the Authorization Services	17
Batch numbers controlled by the merchant	17
Creating new batches – XML explanations	17
Creating new batches – XML examples.....	18
Getting transaction lists.....	19
Getting transaction lists – XML explanations	19
Getting transaction lists – XML examples	21
Getting Disputed Transactions.....	21
Getting disputed transactions.....	23
Getting disputed transaction list – XML examples	24
Getting Fraud Transactions.....	25
Getting fraud transactions	25
Getting fraud transaction list – XML examples	27
Getting Settlement Data	28
Getting settlement list	28
Getting a list of batches in settlement	28
Getting a list of transactions in batches. BatchDate and BatchNumber are unique for each batch.....	29
Getting a list of chargebacks in settlement.....	29
Request:.....	30
Response:.....	30
Request:.....	31
Response:.....	32
Sending detailed data	35
Sending detail data – XML explanations	35
Sending detail data – XML examples.....	39
Sending passenger itinerary.....	40
Sending passenger itinerary – XML explanations.....	40
Sending detail data – XML examples.....	41
Action Codes	42
FAQ.....	43
Frequently Asked Questions	43

About B-Gateway

B-Gateway is a code name for a web service that allows a merchant to obtain authorizations for credit card transactions and to submit transactions for clearing and settlement.

Key features

B-Gateway's key features enable the user to do the following:

Get authorizations – allows the user to either make a *normal purchase transaction* or to pre-authorize it.

The difference is:

If a transaction is *pre-authorized*, the given amount is reserved from the cardholder's account. The amount will not be charged to the cardholder until a financial transaction is submitted. The authorization code needs to be stored by the merchant and later used to send in the financial transaction. Financial transactions are submitted by calling `getAuthorization` with the `AuthCode` from a previous pre-authorized transaction.

With the *normal purchase transaction* the cardholders account is charged for the given amount. If the transaction was pre-authorized then the transaction must include the authorization code received from that transaction.

This feature also allows the user to make a refund transaction if he wants to refund an amount to the cardholder's account.

Cancel authorizations – allows the user to cancel previously authorized transactions. Authorizations can be cancelled if the batch they belong to is still open. If it is closed then a refund transaction must be used.

Create new batches – allows the user to create a new batch. By doing so the user closes the currently open batch and opens a new one.

Get batch authorizations – allows the user to authorize many transactions as one batch.

Get processing results – allows the user to get results from previous batch job.

Get transaction lists – allows the user to get a transaction list from the authorization service listing, on either batch or transaction level details on transactions processed through the service.

Send detail data – allows the user to supply detailed invoice after the financial transaction has been processed.

Introduction to batches

A batch is a unit that the acquirer uses to settle with the merchant. It is a collection of credit card transactions charged over a period of time, usually one day.

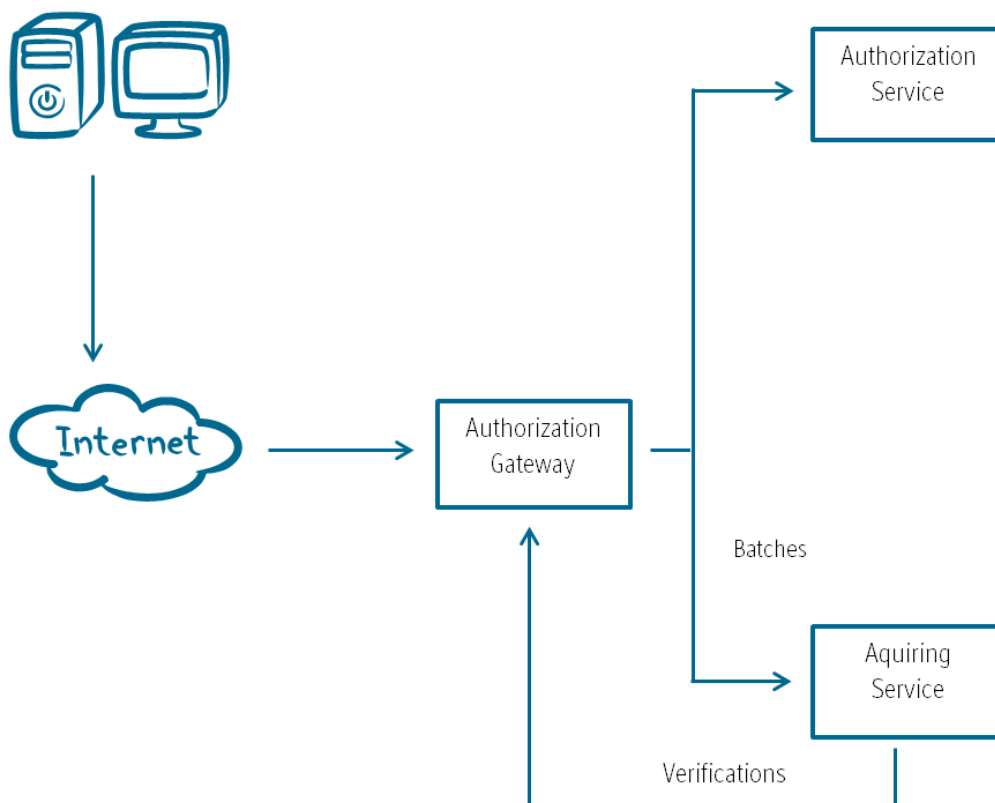
Each batch is identified by a batch number and this batch number must be unique per terminal within one year. The batch number can either be controlled by the merchant or the authorization services. This will be explained later.

How does B-Gateway work?

B-Gateway will expose the authorization services with the use of web services. The web services run securely over HTTPS.

All web services provided by B-Gateway will be password protected. The connecting merchant has to supply username and password each time the merchant calls a B-Gateway web service.

Once the merchant has connected to the web service he will be able to access the features mentioned above. When using these features the merchant must send all necessary information on XML format. This will be explained later. This XML information will then be sent through the web service and a response, also on XML format, will be given.



Model 1 – This diagram shows an example of B-Gateway's process

Data Storage and Manipulation

The security of cardholder account data has become one of the biggest issues facing the payment card industry. Visa and MasterCard have therefore created the Payment Card Industry Data Security Standard (PCI DSS) – a set of industry-wide requirements and processes, that are supported by all international payment card systems.

PCI DSS is intended for any organization that stores, transmits or processes cardholder account and transaction data – merchants, acquiring banks and related service providers.

The PCI standard is available at <https://www.pcisecuritystandards.org> Please download and review the standard.

While all merchants are required to comply with the Payment Card Industry Data Security Standard, merchants that store, process or transmit card account data may also be required to validate compliance with their acquirer. Please note that for a merchant to be considered compliant, any Service Providers that store, process or transmit card account data on behalf of the merchant must also be compliant.

Data types

Field Form

- A Alphabetic
- N Numeric
- AN Alphanumeric
- Z ISO 7811 and ISO 7813 track 2 and track 3 data format

Field Length

- 10 Indicates length=10
- LL..19 Indicates max length of 19
- ..17 Variable length up to 17 characters

Field Code

- M Mandatory
- C Conditional
- O Optional

Amounts

All amounts are represented with two digits after the decimal and an implied decimal point. Icelandic Krona is therefore represented in aurar. Dollars are in cents etc. For instance 100 USD is in TrAmount as “10000”.

Getting authorizations

Normal purchase transaction

This element is referred as TransType 1 in the XML sheet. If you wish to make a normal purchase transaction you can either insert a variable in the AuthCode in the XML sheet or leave the field empty.

The difference is that if the AuthCode is empty B-Gateway will request authorization from the issuer and, if successful, B-Gateway will store the financial transaction in its database and later submit the financial transaction to the acquirer automatically.

OR

If the transaction has already been authorized by the pre-authorizing element or by other channels the AuthCode will contain the auth code that was received during the pre-authorization process. This is sometimes referred to as submitting financial transactions. In this case B-Gateway will store the financial transaction in its database and not request authorization from the issuer to ensure that the card account does not get reserved twice. B-Gateway will then later submit the transaction to the acquirer automatically.

In either case the transaction number and batch number are always returned from the web service. If Transaction and Batch variables are not set in input requests they are created and returned in the response.

Refund

This element is referred as TransType 3 in the XML sheet. This transaction type is used when a merchant wants to refund an amount to a customer. If the refund is successful the response to the merchant contains the AuthCode of the reversal, as well as Transaction and Batch numbers. In other case the response contains an error code.

Partial Reversal

This element is referred as TransType 4 in the XML sheet. This transaction type lowers the amount on a currently existing sale and advises the issuer to change the cardholder's withdrawal limit by the same amount. The transaction type can only be used on a sale **created on the same day and therefore in an open batch**. The message must contain the original amount, and the new amount.

If the partial reversal is successful the response to the merchant contains the AuthCode of the reversal, as well as Transaction and Batch numbers. In other case the response contains an error code.

Pre-authorizations

This element is referred as TransType 5 in the XML sheet. This transaction type is used when the merchant wants to reserve an amount from a customer's credit card. An Authorization Request is then sent to the acquirer. If the request was authorized the transaction is added to a system database containing the new AuthCode. Transaction and batch numbers are not assigned to the transaction as the transaction is only a pre-authorization request. The

response sent to the merchant therefore only contains AuthCode, not Transaction and Batch.

Recurring Zero amount authorization

The Payment Gateway does support zero amount to be sent to verify cards for MasterCard and VISA. The only requirement is that the authorization must be a pre-authorization. This is most often used in recurring sales.

Dynamic Descriptors

DD information is sent by adding the following parameters to the authorization message.

MerchantName	AN 1..22
MerchantHome	AN 1..20
MerchantCity	AN 1..13
MerchantZipCode	AN 1..10
MerchantCountry	A2, two letter country code

DD works with all authorization requests except cancelAuthorization. For pre-authorization (Transtype 5) MerchantName, MerchantCity and MerchantContry will be sent (limited by card schema regulation), for financial authorization (Transtype 1, 3, 4) all parameters are sent, that also applies when pre-authorizations are captured with sale (Transtype 1).

For pre-authorization with capture it is required that the pre-authorization also contains the DD since it is a requirement from VISA, MasterCard uses the financial record.

Ecommerce

It is common in ecommerce to have phonenumber in city. There are two ways to do this with Dynamic Descriptors.

The preferred way is to use additional parameters that define the transactions as ecommerce. MerchantHome and MerchantCity should contain actual address.

Ecommerce	AN 1	Set as 1 (true), 0 or nonexitant (false)
EcommercePhone	AN 1..13	Merchant phonenumber

This can also be done by setting phonenumber in MerchantHome if merchant is marked as ecommerce merchant in Borgum merchant configuration. In this case the actual MerchantHome is stored in Borgum merchant configuration for use in financial transaction sending to scheme.

Restrictions

- Name, ZipCode and City can contain AlphaNumeric letters along with common characters such as *, +, - _ . While no characters are explicitly forbidden then the reality is that several issuer dislike uses of characters such as &, <, >, ?, /, . \ and they should be avoided.
- The merchant contract numbers have to be specifically configured through business channels to allow use of dynamic descriptors, if not then these parameters will be ignored.
- MerchantName must always have value and the proper way is to fill in all the parameters. If a parameter is missing then the gateway will fill out the rest with pre-configured values which might not properly represent the merchant. Exception to

this is MasterCard which has restrictions that requires home, zipcode and country to refer to an actual address.

See example below.

Getting authorizations – XML explanations

Table GA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is “1000”
Processor	7	N	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	..8	AN	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalID	..8		<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services. Must be set if default terminal is not defined for merchant.
TransType	1	N	<i>Mandatory</i>	1 = Normal purchase transaction 3 = Refund 4 = Partial Reversal 5 = Pre-Authorize
TrAmount	..12		<i>Mandatory</i>	Amount, in general it includes two decimal points, i.e. 100 USD is 10000. Currently JPY is the only exception, it contains no decimal points.
NewAmount	..12		<i>Mandatory</i>	Only used by transtype 4. Contains the new amount to override the original sale amount.
TrCurrency	3	N	<i>Mandatory</i>	Transaction currency ISO 4217 352 => ISK 978 => EUR
DateAndTime	12	N	<i>Mandatory</i>	YYMMDDhhmmss. When repeated use original date and time
PAN	LL..19	N	<i>Conditional</i>	Card number – Used if a card number is entered manually. See also Track1/Track2. Card number can be a virtual card (see function getVirtualCard). If TransType is refund then PAN can be empty, PAN from a transaction of TransType=1 with the same DateAndTime + RRN will be used.
ExpDate	4	N	<i>Conditional</i>	Expiration date – YYMM not used if Track1/Track2
Track1	LL..6	AN	<i>Conditional</i>	Mandatory if magnetic stripe read and not Track2
Track2	LL--37	Z	<i>Conditional</i>	Mandatory if magnetic stripe read
RRN	12	AN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction. Must be exactly 12 characters . Recommended format is a fixed value followed by a sequence, for example ACME00012345. The last six letters must contain a numeric value.
AuthCode	..6	AN	<i>Conditional</i>	Authorization code of a previous pre-authorized transaction. Include this code to use the previously gotten authorization code
CVC2	3	N	<i>Optional</i>	Used in cardholder not present situations such as internet sales. Code is printed on the back of the cardholder’s card

SecurityLevelInd	1	AN	<i>Optional</i>	<p>Security Level Indicator – Used with 3dsecure to indicate the security level used in electronic transactions</p> <p>This property is mandatory in electronic commerce and is used to identify the level of security used when doing the transaction.</p> <p>0 = 3dsecure not supported by merchant 1 = 3dsecure supported by merchant but not provided by issuer 2 = 3dsecure is present</p> <p>Mapping from Borgun MPI is MPI(1) → 2 MPI(2,3,4) → 1 MPI(5,6,7,8) → 1</p> <p>Mapping from VISA ECI 5 → 2 ECI 6 → 1 All else → 0</p> <p>MasterCard needs SecurityLevel + UCAF VISA needs SecurityLevel + CAVV + XID</p> <p>If sent in as pre-authorization then the same attributes must be in capture.</p>
UCAF	32	AN	<i>Conditional</i>	Universal Cardholder Authentication Field – Used with Secure Code
CAVV	28	AN	<i>Conditional</i>	Verified by Visa, Base64 encoded cardholder authentication verification value, f.ex AAACCRSQJ4lShoBncZAnAAAAA=
XID	28	AN	<i>Conditional</i>	Verified by Visa. The unique transaction action id identifier encode by Base64.
MerchantName	1..22	AN	<i>Conditional</i>	Name of merchant.
MerchantHome	1..20	AN	<i>Conditional</i>	Merchant home
MerchantCity	1..13	AN	<i>Conditional</i>	City of merchant or contact information (f.ex. The Shop, +123456789)
MerchantZipCode	1..10	AN	<i>Conditional</i>	ZipCode of merchant shop
MerchantCountry	2	A	<i>Conditional</i>	Two letter country code in uppercase (f.ex. GB, US)
Ecommerce	1	A	<i>Conditional</i>	If set to 1 then transaction is marked as ecommerce, has effect on EcommercePhone .
EcommercePhone	1..13	A	<i>Conditional</i>	If Ecommerce is set then this phonenumber will follow the transaction.

Refund and Partial Refunds must refer to the original transaction by having exactly the same DateAndTime and RRN as the original authorization.

Table GA.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is “1000”
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalID	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services.
TransType	<i>Echo</i>	1 = Normal purchase transaction 3 = Refund 5 = Pre-Authorize
TrAmount	<i>Echo</i>	Amount, note for example 100 USD is 10000 because the amount is represented in cents
TrCurrency	<i>Echo</i>	Transaction currency ISO 4217 352 => ISK 978 => EUR
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. When repeated use original date and time
PAN	<i>Conditional</i>	Card number – Used if a card number is entered manually. See also Track1/Track2
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction.
Transaction	<i>Conditional</i>	Unique number of the transaction within batch.
Batch	<i>Conditional</i>	Unique number of the batch in use for this terminal.
CVCResult	<i>Conditional</i>	If available indicates how the issuer processed the CVC. Values: M = Valid CVC N = Invalid CVC P = CVC not processed (issuer unavailable)
CardAcclid	<i>Mandatory</i>	Card Acceptor Identification Code Code Identifying the card acceptor. This Id is issued by the acquirer
CardAccTerminal	<i>Optional</i>	Card Acceptor Terminal Identification Code Code Identifying the terminal the card acceptor is using
CardAccName	<i>Mandatory</i>	Card Acceptor Name and Location Name\Street\City\Postal Code\Region (two digit US state code or three digit ISO alpha country code if outside US)\Country Code as in ISO3166
AuthCode	<i>Mandatory</i>	Authorization code if the transaction was accepted
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	A numeric code that identifies the store and terminal within each merchant chain.
CardType	<i>Mandatory</i>	If available this will indicate the name of the card type
Message	<i>Optional</i>	Contains a text message if form error occurs

Getting authorizations – XML examples

GAXML.1 – An example of authorization request

```
<?xml version='1.0' encoding='utf-8'?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>5</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401711111116716</PAN>
  <ExpDate>0705</ExpDate>
  <RRN>TESTING00001</RRN>
  <CVC2>123</CVC2>
</getAuthorization>
```

GAXML.2 – An example of an authorization capture

```
<?xml version='1.0' encoding='utf-8'?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>1</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <RRN>TESTING00001</RRN>
  <AuthCode>060215012345</ AuthCode >
</getAuthorization>
```

GAXML.3 – An example of refund of sale

```
<?xml version='1.0' encoding='utf-8'?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>1</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>3</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <RRN>TESTING00001</RRN>
</getAuthorization>
```

GAXML.4 – An example of partial reversal of sale

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>1</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>4</TransType>
  <TrAmount>100000</TrAmount>
  <NewAmount>30000</NewAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <RRN>TESTING00001</RRN>
</getAuthorization>
```

GAXML.1 – An example of authorization request with Dynamic Descriptor

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>2</Processor>
  <MerchantID>2</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>1</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401711111116716</PAN>
  <ExpDate>0705</ExpDate>
  <RRN>TESTING00001</RRN>
  <CVC2>123</CVC2>
  <MerchantName>The Shop</MerchantName>
  <MerchantHome>The Road 123</MerchantHome>
  <MerchantCity>The City</MerchantCity>
  <MerchantZipCode>AL123</MerchantZipCode>
  <MerchantCountry>UK</MerchantCountry>
  <Ecommerce>1</Ecommerce>
  <EcommercePhoneNumber>>+123456789</ EcommercePhoneNumber >
</getAuthorization>
```

Canceling authorizations

This operation cancels a previously authorized transaction. The previously authorized transaction is found by searching for the AuthCode of the transaction. If no AuthCode is present in a request, the most recent transaction that matches the criteria (i.e. the rest of the input request) is cancelled. Transactions authorized with TransType=1 can only be cancelled while the batch they belong to is still open. Transactions authorized with TransType=5 can always be cancelled as they don't belong to any batches.

The card number and amount must be the same as in the original transaction. In the event of an error, such as timeout in communications, the RRN number of the transaction in the error must be included in the cancelling process.

Authorizations can be cancelled if the batch they belong to is still open. If it is closed then a refund transaction must be used. Pre-Authorize transaction can be cancelled after the batches are closed since pre-authorized transactions do not belong to batches.

Canceling authorizations – XML explanations

Table CA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is “1000”
Processor	7	N	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	..8	AN	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalID	..8	AN	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services. Must be set if default terminal is not defined for merchant.
TransType	1	N	<i>Mandatory</i>	Same as the original transaction
TrAmount	..12	N	<i>Mandatory</i>	Same as the original transaction
TrCurrency	3	N	<i>Mandatory</i>	Same as the original transaction
DateAndTime	12	N	<i>Mandatory</i>	YYMMDDhhmmss. Same as the original transaction
PAN	LL..19	N	<i>Conditional</i>	Card number – Same as the original transaction. If PAN is empty then DateAndTime + RRN will be used to find PAN in original transaction.
Track1	LL..76	AN	<i>Conditional</i>	Same as the original transaction
Track2	LL..37	Z	<i>Conditional</i>	Same as the original transaction
RRN	12	AN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction. Same as the original transaction
AuthCode	..6	AN	<i>Conditional</i>	Same as the original transaction

Table CA.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is “1000”
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalID	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services. Must be set if default terminal is not defined for merchant.
TransType	<i>Echo</i>	Same as the original transaction
TrAmount	<i>Echo</i>	Same as the original transaction
TrCurrency	<i>Echo</i>	Same as the original transaction
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. Same as the original transaction
PAN	<i>Conditional</i>	Card number – Same as the original transaction
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction. Same as the original transaction
Transaction	<i>Conditional</i>	Number of the transaction. Same as the original transaction
Batch	<i>Conditional</i>	Number of the batch in use. Same as the original transaction
CardAcclid	<i>Mandatory</i>	Code Identifying the card acceptor. Same as the original transaction
CardAccTerminal	<i>Mandatory</i>	Same as the original transaction

CardAccName	<i>Mandatory</i>	Card Acceptor Name and Location Name\Street\City\Postal Code\Region (two digit US state code or three digit ISO alpha country code if outside US)\Country Code as in ISO3166
AuthCode	<i>Mandatory</i>	Authorization code if the transaction was accepted
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	A numeric code that identifies the store and terminal within each merchant chain.
CardType	<i>Mandatory</i>	If available this will indicate the name of the card type
Message	<i>Optional</i>	A message in comprehension with the Action Code

Cancelling authorizations – XML examples

CAXML.1 – An example of sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<cancelAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalId>1</TerminalId>
  <TransType>1</TransType>
  <TrAmount>000000100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401710000006716</PAN>
  <ExpDate>1405</ExpDate>
  <RRN>060215012345</RRN>
  <AuthCode>809285</AuthCode>
</cancelAuthorization>
```

Virtual card numbers

Credit card numbers need not be sent into getAuthorization, instead virtual card numbers can be used. This enhances security as it eliminates the need to store sensitive credit card information in local data sources.

Virtual card numbers – XML explanations

Table VA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
MerchantContractNumber	7	AN	<i>Mandatory</i>	Contract number issued by Borgun.
PAN	16	N	<i>Mandatory</i>	Card number

Table VA.2 – Field explanations for a received XML

Field	Type	Explanation
VirtualCard	<i>Mandatory</i>	Card number that can be used as reference to the real card number in authorization.

Cancelling authorizations – XML examples

VCXML.1 – An example of sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<getVirtualCard>
  <MerchantContractNumber>7891234</MerchantContractNumber>
  <PAN>540122233334444</PAN>
</ getVirtualCard >
```

Creating new batches

New batch closes the currently open batch and opens a new one.

If the currently open batch has no transactions associated with it, the batch will be considered as being the new batch, thus it will not be closed nor will a new batch be opened.

Batch numbers controlled by the Authorization Services

This is the preferred way of handling batches. The merchant sends in authorization requests to the Authorization Services. The Authorization Services will add the batch and transaction numbers to the transaction as needed. When the merchant wants to close the batch the merchant can use the NewBatch function or the Authorization Services can take care of this daily at predefined time.

Batch numbers controlled by the merchant

The merchant adds the batch and transaction numbers to the authorization request. In order to start a new batch the merchant simply increments the batch number and starts the transaction number back at number 1. The batch number must be unique for particular terminal within the year and in ascending sequence. Transaction numbers must be unique within a batch number.

Creating new batches – XML explanations

Table CNB.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	7	N	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	..8	AN	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services

TerminalId	..8	AN	<i>Optional</i>	If not set then the batch for the default terminal will be closed. If set then the insert terminal will be close.
DateAndTime	12	AN	<i>Optional</i>	YYMMDDhhmmss. Only used for documentation purposes

Table CNB.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalId	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. Same as the request
NewBatch	<i>Conditional</i>	Number of the new batch in use
OldBatch	<i>Conditional</i>	Number of the batch just closed
CardAccTerminal	<i>Optional</i>	Terminal id
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	If available this will indicate the name of the card type

Creating new batches – XML examples

CNBXML.1 – An example of a sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<newBatch>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <TerminalId>1</TerminalId>
  <DateAndTime>060216103700</DateAndTime>
</newBatch>
```

Getting transaction lists

Client applications can request a transaction list from the authorization service listing on either batch or transaction level details about transactions processed through the service.

It is possible to specify a batch number in the request to get all transactions that belong to that given batch. It is also possible to get a list of transactions for a given time period.

Getting transaction lists – XML explanations

Table TL.1 – Field explanations for a sent XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalId	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services. Must be set if default terminal is not defined for merchant.
BatchNumber	<i>Conditional</i>	Batch number. If the BatchNumber is omitted, the response will only include batchnumbers processed during the given time period. Otherwise a list of transactions for the given batchnumber is returned.
FromDate	<i>Conditional</i>	Format:YYYYMMDD. Mandatory field if BatchNumber is empty.
ToDate	<i>Conditional</i>	Format:YYYYMMDD. Only used if BatchNumber is empty. If this parameter is not set, current date will be used as ToDate.
RRN	<i>Conditional</i>	Searches all batches for transactions with the given RRN

Tables TL.2 and TL.3 – Field explanations for a received XML

Explanations for the header of the XML output

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
ActionCode	<i>Mandatory</i>	Reason code for response

Explanations for each transaction within the batch

Field	Type	Explanation
TransType	<i>Mandatory</i>	1 = Normal purchase transaction 3 = Refund 5 = Pre-Authorize
TransactionNumber	<i>Mandatory</i>	Number of the transaction
BatchNumber	<i>Mandatory</i>	Number of the requested batch
TransactionDate	<i>Mandatory</i>	YYMMDDhhmmss. Date and time of the transaction
PAN	<i>Mandatory</i>	Card number – Used if a card number is entered manually
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction
ActionCode	<i>Conditional</i>	Reason code for response
AuthorizationCode	<i>Mandatory</i>	Authorization code of a previous pre-authorized transaction. Include this code to use the previously gotten authorization code
TrAmount	<i>Mandatory</i>	Amount, note for example 100 USD is 10000 because the amount is represented in cents
TrCurrency	<i>Mandatory</i>	Transaction currency ISO 4217 352 => ISK 978 => EUR
CardType	<i>Conditional</i>	If available this will indicate the name of the card type
Voided	<i>Mandatory</i>	Voided can have values T (True) and V (Void transactions). A transaction that is voided will get Voided as T, the voiding transaction (cancelAuthorization) will get voided as V. The void status can only be created by cancelAuthorization().
Status	<i>Mandatory</i>	Status on transaction can have the following values 1: Sale initial status 2: Sale successful 3: Sale failed 4: Refund initial status 5: Refund successful 6: Refund failed 7: Pre-authorization initial status 8: Pre-authorization successful 9: Pre-authorization failed 10: Reversal initial status 11: Reversal successful 12: Reversal failed 14: Financial record sent A successful transaction will be in state 2 (or 5 for refund) and end up in state 14 when the financial record has been sent to the card schema. Other statuses are more for information purposes, i.e.

		have no financial impact.
TerminalNr		Id of terminal being used during authorization
Credit	<i>Conditional</i>	Contains 1 if OCT/CFT transaction

Getting transaction lists – XML examples

TLXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<TransactionListRequest>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <BatchNumber>3</BatchNumber>
</TransactionListRequest>
```

Getting Disputed Transactions

Client applications can request a list of disputed transactions.

All authorization transactions have a unique number named Acquiring Reference, for each transaction that has a chargeback there will be a list (1 or more) of chargeback records detailing the history of the chargeback. They can be fetched by using the Acquiring Reference.

Dispute types returned are as follows

- **Chargeback**
 - Issuer to acquirer. Chargeback of the original transaction, short code is 'C'. If the chargeback is partial, that is only partial amount of the original transaction is used, then the short code is 'PC'
- **Retrieval Request / Information Retrieval**
 - Issuer to acquirer. Request for information regarding the original transaction, non financial impact, short code is 'RR'.
- **Chargeback Reversal**
 - Issuer to acquirer. Reverses the chargeback, short code is 'CR'.
- **Second Presentment / Re-Presentment**
 - Acquirer to Issuer. Negates the chargeback by sending the financial transaction again, initiated by merchant, short code is 'SP'.
 - If the re-presentment is partial, that is only partial amount of the original transaction is used, then the short code is 'PSP'
- **Second Chargeback / Arbitration Chargeback**
 - Issuer to acquirer. Chargeback in response to the re-presentment, short code is 'AC'.

If the arbitration chargeback is partial, that is only partial amount of the original transaction is used, then the short code is '**PAC**'

– **Second Presentment Reversal**

- Acquirer to Issuer. Negates a Second Presentment / Re-Presentment, can be caused by second presentment being denied by schema or manual mistakes. Rarely used. short code is '**SPR**'.

The chargeback list describes the communication between acquirer and issuer during dispute, a path for a long running dispute might be creating the following four records

1. Retrieval Request / Information Retrieval: Issuer needs more information to determine if the original transaction should be accepted or a chargeback sent.
2. Chargeback: Issuer requests a chargeback from acquirer. Issuer can always back out of the chargeback by doing a chargeback reversal.
3. Second Presentment / Re-Presentment: Acquirer does not accept the chargeback and resends the transaction.
4. Second Presentment Reversal: Acquirer negates a second presentment.
5. Issuer might then end with Second Chargeback(Issuer re-request for chargeback)

A rare occurrence that still happens once in a while is double registration of chargeback.

1. Chargeback: Issuer requests a chargeback from acquirer
2. Chargeback: A second chargeback from issuer to acquirer
3. Chargeback Reversal: Issuer reverses one chargeback for acquirer does a re-presentment.

Getting disputed transactions

Table DT.1 – Parameters for `getDisputedTransactions ()`

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
DateFrom	<i>Mandatory</i>	Returns all disputed transactions from date DateFrom. Date format YYYYMMDD.
DateTo	<i>Mandatory</i>	Returns all disputed transactions up to and including date DateFrom. Date format YYYYMMDD.

Table DT.2 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	<i>Mandatory</i>	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	<i>Mandatory</i>	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	<i>Mandatory</i>	Detailed error description
MessageId	<i>Mandatory</i>	Error description code of ErrorMessage if applicable
ReferenceValue	<i>Mandatory</i>	Id of record within list where error occurred. Only returned if applicable
StackTrace	<i>Mandatory</i>	Stacktrace of error if applicable

Table DT.3 – Field explanations for a received XML

Field	Type	Explanation
MerchantNumber	<i>Mandatory</i>	Merchant contract number
ProcessingTime	<i>Mandatory</i>	Date and time of record processing at Borgun, yyyy-mm-ddThh:mm:ss
Date	<i>Mandatory</i>	Date of record creation, yyyy-mm-dd
Brand	<i>Mandatory</i>	Card brand type
Type	<i>Mandatory</i>	See table above
Amount	<i>Mandatory</i>	Current amount
Currency	<i>Mandatory</i>	Current currency
ReasonCode	<i>Mandatory</i>	Code describing reason for dispute
ReasonText	<i>Mandatory</i>	Text describing reason for dispute
AcquiringReferenceData	<i>Mandatory</i>	Acquirer reference
BatchDate	<i>Mandatory</i>	Batch date of transaction, yyyy-mm-dd
TerminalId	<i>Mandatory</i>	Transaction terminal id
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction
CardNumber	<i>Mandatory</i>	Masked cardnumber in transaction
OriginalAmount	<i>Mandatory</i>	Amount in transaction
OriginalCurrency	<i>Mandatory</i>	Currency of transaction
OriginalDate	<i>Mandatory</i>	Date of transaction, yyyy-mm-dd
AuthorizationCode	<i>Mandatory</i>	Transaction authorization number

Getting disputed transaction list – XML examples

DTXML.1 – An example of a sent and received XML

```
<?xml version="1.0" encoding="utf-8"?>
<getDisputedTransactions>
  <Processor>21</Processor>
  <MerchantContractNumber>9275444</MerchantContractNumber>
  <DateFrom>20101021</DateFrom>
  <DateTo>20111025</DateTo>
</getDisputedTransactions>
```

```
<?xml version="1.0"?>
<DisputedTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <DisputedTransactionList>
    <DisputedTransaction>
      <MerchantNumber>9275444</MerchantNumber>
      <ProcessingTime>2008-09-18T00:00:00</ProcessingTime>
      <Date>2008-09-18</Date>
      <Brand>MC</Brand>
      <Type>C</Type>
      <Amount>2000000</Amount>
      <Currency>352</Currency>
      <ReasonCode>4809</ReasonCode>
      <ReasonText>TRANSACTION NOT RECONCILED</ReasonText>
      <AcquiringReferenceData>5414838259002000123123</AcquiringReferenceData>
      <BatchDate>2008-06-23</BatchDate>
      <TerminalId>00010002</TerminalId>
      <RRN>111222</RRN>
      <CardNumber>002359*****3123</CardNumber>
      <OriginalAmount>20000</OriginalAmount>
      <OriginalCurrency>352</OriginalCurrency>
      <OriginalDate>2008-06-21</OriginalDate>
      <AuthorizationCode>0</AuthorizationCode>
    </DisputedTransaction>
  </DisputedTransactionList>
</DisputedTransactionResponse>
```


Getting Fraud Transactions

Client applications can request a list of fraud transactions.

Getting fraud transactions

Table FT.1 – Parameters for getFraudTransactions ()

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
DateFrom	<i>Mandatory</i>	Returns all fraud transactions from date DateFrom. Date format YYYYMMDD.
DateTo	<i>Mandatory</i>	Returns all disputed transactions up to and including date DateFrom. Date format YYYYMMDD.

Table FT.2 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	<i>Mandatory</i>	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	<i>Mandatory</i>	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	<i>Mandatory</i>	Detailed error description
MessageId	<i>Mandatory</i>	Error description code of ErrorMessage if applicable
ReferenceValue	<i>Mandatory</i>	Id of record within list where error occurred. Only returned if applicable
StackTrace	<i>Mandatory</i>	Stacktrace of error if applicable

Table DT.3 – Field explanations for a received XML

Field	Type	Explanation
MerchantNumber	<i>Mandatory</i>	Merchant contract number
ProcessingTime	<i>Mandatory</i>	Date and time of record processing at Borgun, yyyy-mm-ddThh:mm:ss
Date	<i>Mandatory</i>	Date of record creation, yyyy-mm-dd
Brand	<i>Mandatory</i>	Card brand type
Type	<i>Mandatory</i>	Currently can only have the value 'F' (Fraud)
ReasonCode	<i>Mandatory</i>	Code describing reason for dispute
ReasonText	<i>Mandatory</i>	Text describing reason for dispute
AcquiringReferenceData	<i>Mandatory</i>	Acquirer reference to the original authorization transaction that this dispute record relates to. For each disputed transaction there exists 1 or more dispute record that all will have the same acquirer reference.
NotificationCode	<i>Mandatory</i>	Notification code 1 = Addition 2 = Addition of subsequent duplicate transaction 3 = Change 4 = Delete 5 = Reactivate
IssuerGeneratedAuthorization	<i>Mandatory</i>	VISA specific, Y = Issuer authorized transaction X = Transaction authorized but not by issuer N = Transaction not authorized
AccountSequenceNumber	<i>Mandatory</i>	VISA specific, identifies a transaction within the account number. A maximum of 1000 transactions on a specific account may be added to the Visa fraud master file. The member may also pre-assign a sequence number within the 4000–4999 range. account
FraudInvestigationStatus	<i>Mandatory</i>	VISA specific, member-assigned, may be 00 through 99 or AA through ZZ
CustomerReference	<i>Mandatory</i>	VISA specific
BatchDate	<i>Mandatory</i>	Batch date, yyyy-mm-dd
TerminalId	<i>Mandatory</i>	Transaction terminal id
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction
CardNumber	<i>Mandatory</i>	Masked cardnumber in transaction
OriginalBatchId	<i>Mandatory</i>	VISA specific, original batch id
OriginalAmount	<i>Mandatory</i>	Amount in transaction
OriginalCurrency	<i>Mandatory</i>	Currency of transaction
OriginalDate	<i>Mandatory</i>	Date of transaction, yyyy-mm-dd
AuthorizationCode	<i>Mandatory</i>	Transaction authorization number

Getting fraud transaction list – XML examples

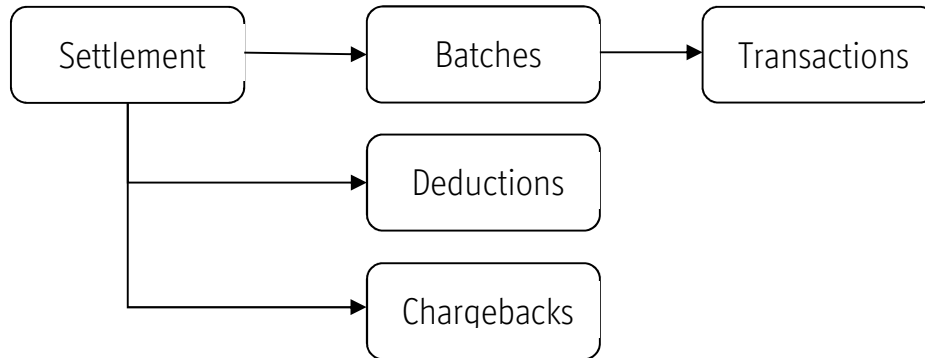
DTXML.1 – An example of a received XML

```
<?xml version="1.0"?>
<FraudTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <FraudTransactionList>
    <FraudTransaction>
      <MerchantNumber>9577851</MerchantNumber>
      <ProcessingTime>2009-10-14T10:25:00.60831</ProcessingTime>
      <Date>2009-03-12</Date>
      <Brand>MC</Brand>
      <Type>F</Type>
      <ReasonCode>4</ReasonCode>
      <ReasonText>COUNTERFEIT</ReasonText>
      <AcquiringReferenceData>5414838196002000012345</AcquiringReferenceData>
      <NotificationCode>1</NotificationCode>
      <IssuerGeneratedAuthorization></IssuerGeneratedAuthorization>
      <AccountSequenceNumber></AccountSequenceNumber>
      <FraudInvestigationStatus></FraudInvestigationStatus>
      <CustomerReference></CustomerReference>
      <BatchDate>2008-06-23</BatchDate>
      <TerminalId>00010002</TerminalId>
      <RRN></RRN>
      <CardNumber>398320*****2345</CardNumber>
      <OriginalBatchId>1</OriginalBatchId>
      <OriginalAmount>10000</OriginalAmount>
      <OriginalCurrency>352</OriginalCurrency>
      <OriginalDate>2008-06-21</OriginalDate>
      <AuthorizationCode>0</AuthorizationCode>
    </FraudTransaction>
  </FraudTransactionList>
</FraudTransactionResponse>
```

Getting Settlement Data

Client applications can fetch settlement data.

Settlement can contain one or more batches, and each batch contains one or more transaction.



Settlement data can be fetched with the following calls

Getting settlement list

Table SD.1 – Parameters for `getSettlement ()`

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
DateFrom	<i>Mandatory</i>	Returns all settlements from date DateFrom. Date format YYYYMMDD.
DateTo	<i>Mandatory</i>	Returns all settlement up to and including DateTo. Date format YYYYMMDD.

Getting a list of batches in settlement

Table SD.2 – Parameters for `getSettlementBatch()`

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
SettlementNumber	<i>Mandatory</i>	Unique number of settlement

Getting a list of transactions in batches. BatchDate and BatchNumber are unique for each batch.

Table SD.3 – Parameters for getSettlementTransactions()

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
BatchNumber	<i>Mandatory</i>	Batch number
BatchDate	<i>Mandatory</i>	Batch date, format YYYYMMDD

Getting a list of chargebacks in settlement

Table SD.4 – Parameters for getSettlementChargeback()

Field	Type	Explanation
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	<i>Mandatory</i>	Contract number issued by Borgun.
SettlementNumber	<i>Mandatory</i>	Unique number of settlement

Table SD.4 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	<i>Mandatory</i>	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	<i>Mandatory</i>	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	<i>Mandatory</i>	Detailed error description
MessageId	<i>Mandatory</i>	Error description code of ErrorMessage if applicable
ReferenceValue	<i>Mandatory</i>	Id of record within list where error occurred. Only returned if applicable
StackTrace	<i>Mandatory</i>	Stacktrace of error if applicable

Tables SD.5 – Settlement

Field	Type	Explanation
SettlementNumber	<i>Mandatory</i>	Settlement number
SettlementDate	<i>Mandatory</i>	Date of settlement
MastercardDomesticAmount	<i>Conditional</i>	Not in use
MastercardInternationalAmount	<i>Conditional</i>	Not in use
MastercardOtherAmount	<i>Conditional</i>	Not in use
JcbAmount	<i>Conditional</i>	Not in use
VisaAmount	<i>Conditional</i>	Not in use
AmexAmount	<i>Conditional</i>	Not in use
TotalAmount	<i>Mandatory</i>	Total amount in settlement
TransactionCountInSettlement	<i>Mandatory</i>	Number of transactions in settlement
SettlementCurrencyCode	<i>Mandatory</i>	Currency code of settlement
DeductionAmount	<i>Mandatory</i>	Deduction amount of settlement amount
RefundAmount	<i>Mandatory</i>	Refund amount in settlement

Tables SD.6 – Settlement deductions

Field	Type	Explanation
SettlementNumber	<i>Mandatory</i>	Settlement number
Code	<i>Mandatory</i>	Code describing deduction type
CodeDescription	<i>Mandatory</i>	Text describing deduction type
Amount	<i>Mandatory</i>	Deduction amount
DeductionCurrencyCode	<i>Mandatory</i>	Currency code of deduction

SDXML.5 and SDXML6 – An example of a received XML with settlement and deduction

Request:

```
<?xml version="1.0" encoding="utf-8"?>  
<getSettlements>  
  <Processor>1</Processor>  
  <MerchantContractNumber>9999999</MerchantContractNumber>  
  <DateFrom>20110129</DateFrom>  
  <DateTo>20110930</DateTo>  
</getSettlements>
```

Response:

```

<?xml version="1.0"?>
<SettlementResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <Settlement>
    <MerchantNumber>9999999</MerchantNumber>
    <SettlementNumber>4290310</SettlementNumber>
    <SettlementDate>2014-06-04T00:00:00</SettlementDate>
    <MastercardDomesticAmount>0.0</MastercardDomesticAmount>
    <MastercardInternationalAmount>0.0</MastercardInternationalAmount>
    <MastercardOtherAmount>0</MastercardOtherAmount>
    <JcbAmount>0</JcbAmount>
    <VisaAmount>0</VisaAmount>
    <AmexAmount>0</AmexAmount>
    <TotalAmount>220</TotalAmount>
    <TransactionCountInSettlement>0</TransactionCountInSettlement>
    <SettlementCurrencyCode>USD</SettlementCurrencyCode>
    <DeductionAmount>197.62</DeductionAmount>
    <RefundAmount>0.0</RefundAmount>
    <SettlementDeductions>
      <SettlementNumber>4290310</SettlementNumber>
      <Code>F_MSC_MC</Code>
      <CodeDescription>Merchant Commission MC</CodeDescription>
      <Amount>-21.04</Amount>
      <DeductionCurrencyCode>USD</DeductionCurrencyCode>
    </SettlementDeductions>
    <SettlementDeductions>
      <SettlementNumber>4290310</SettlementNumber>
      <Code>F_CBK</Code>
      <CodeDescription>Chargeback Fee</CodeDescription>
      <Amount>-30</Amount>
      <DeductionCurrencyCode>USD</DeductionCurrencyCode>
    </SettlementDeductions>
  </Settlement>
</SettlementResponse>

```

SD.7 – List of batches in settlement

Field	Type	Explanation
BatchNumber	<i>Mandatory</i>	Batch number
BatchDate	<i>Mandatory</i>	Date of batch
SettlementDate	<i>Mandatory</i>	Settlement date
SettlementNumber	<i>Mandatory</i>	Settlement number
TransactionCountInBatch	<i>Mandatory</i>	Number of transactions in batch
TransactionSumInBatch	<i>Mandatory</i>	Amount in transactions in batch
BatchCurrencyCode	<i>Mandatory</i>	Currency code of amounts

SDXML.7 – Example of received batchlist

Request:

```

<?xml version="1.0" encoding="utf-8"?>
<getSettlementBatch>
  <Processor>1</Processor>

```

```
<MerchantContractNumber>9999999</MerchantContractNumber>  
<SettlementNumber>4245</SettlementNumber>  
</getSettlementBatch>
```

Response:

```
<?xml version="1.0"?>  
<SettlementBatchResponse>  
  <Status>  
    <ResultCode>0</ResultCode>  
    <ResultText>Success</ResultText>  
  </Status>  
  <SettlementBatch>  
    <BatchNumber>116757</BatchNumber>  
    <BatchDate>2007-12-05T00:00:00</BatchDate>  
    <SettlementDate>2008-01-02T00:00:00</SettlementDate>  
    <SettlementNumber>4245</SettlementNumber>  
    <TransactionCountInBatch>9</TransactionCountInBatch>  
    <TransactionSumInBatch>18405.00</TransactionSumInBatch>  
    <BatchCurrencyCode>ISK</BatchCurrencyCode>  
  </SettlementBatch>  
</SettlementBatchResponse>
```


SD.8 – Transactions in batches

Field	Type	Explanation
TransactionDate	Mandatory	Date of transaction
Amount	Mandatory	Transaction amount
CurrencyCode	Mandatory	Currency code of amounts in transaction
SlipNumber	Mandatory	Slip number
RRN	Mandatory	Retrieval reference number
CardNumber	Mandatory	Card number
CardType	Mandatory	Card type
OriginalAmount	Mandatory	Original amount on slip
OriginalAmountCurrencyCode	Mandatory	Amount in currency code
BatchNumber	Mandatory	Number of batch containing transaction
SettlementDate	Mandatory	Date of settlement
SettlementNumber	Mandatory	Number of settlement containing transaction

SDXML.8 – An XML example of a received transaction list

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<getSettlementTransactions>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <BatchNumber>101166</BatchNumber>
  <BatchDate>20110105</BatchDate>
</getSettlementTransactions>
```

Response:

```
<?xml version="1.0"?>
<SettlementTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <SettlementTransaction>
    <TransactionDate>2011-01-05T00:00:00</TransactionDate>
    <Amount>687.00</Amount>
    <CurrencyCode>ISK</CurrencyCode>
    <Slipnumber>9372</Slipnumber>
    <RRN>rr1402530644</RRN>
    <Cardnumber>544602*****8005</Cardnumber>
    <Cardtype>MD</Cardtype>
    <OriginalAmount>687.00</OriginalAmount>
    <OriginalAmountCurrencyCode>SEK</OriginalAmountCurrencyCode>
    <BatchNumber>101166</BatchNumber>
    <SettlementDate>2008-01-02T00:00:00</SettlementDate>
    <SettlementNumber>2727</SettlementNumber>
  </SettlementTransaction>
</SettlementTransactionResponse>
```

SD.9 – Chargeback transactions in settlements

Field	Type	Explanation
TransactionDate	Mandatory	Date of transaction
Amount	Mandatory	Transaction amount
CurrencyCode	Mandatory	Currency code of amounts in transaction
SlipNumber	Mandatory	Slip number
RRN	Mandatory	Retrieval reference number
CardNumber	Mandatory	Card number
CardType	Mandatory	Card type
OriginalAmount	Mandatory	Original amount on slip
OriginalAmountCurrencyCode	Mandatory	Amount in currency code
BatchNumber	Mandatory	Number of batch containing transaction
SettlementDate	Mandatory	Date of settlement
SettlementNumber	Mandatory	Number of settlement containing transaction
AcquirerReference	Mandatory	Acquirer reference for chargeback. Unique number for the authorization transaction. Can be used to reference a list of dispute records related to the chargeback.
ChargebackDate	Mandatory	Date of chargeback registration
ChargebackAmount	Mandatory	Amount of chargeback

SDXML.9 – An XML example of a received chargeback list

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<getSettlementChargeback>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <SettlementNumber>7566</SettlementNumber>
</getSettlementChargeback>
```

Response:

```
<?xml version="1.0"?>
<SettlementChargebackResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>OK</ResultText>
  </Status>
  <SettlementChargeback>
    <SettlementDate>2011-08-10T00:00:00</SettlementDate>
    <SettlementNumber>7566</SettlementNumber>
    <BatchNumber>162</BatchNumber>
    <Amount>-34.9500000000000000</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <Cardnumber>4060 42** **** 8831</Cardnumber>
    <Cardtype>04</Cardtype>
    <TransactionDate>2011-07-08T00:00:00</TransactionDate>
    <OriginalAmountCurrencyCode>USD</OriginalAmountCurrencyCode>
    <OriginalAmount>-34.95</OriginalAmount>
    <Slipnumber>777003</Slipnumber>
    <RRN>AA1234567890</RRN>
    <AcquirerReference>541481112222000025737</AcquirerReference>
    <ChargebackDate>2011-08-03T00:00:00</ChargebackDate>
    <ChargebackAmount>187</ChargebackAmount> </SettlementChargeback>
</SettlementChargebackResponse>
```

Sending detailed data

This addition to get Authorization is used to supply detailed invoice after the financial transaction has been processed. In order to match the detail transaction to the financial transaction some parameters must be supplied.

Sending detail data – XML explanations

Tables SDD.1, SDD.2 and SDD.3 – Field explanations for a sent XML

The input is split into a header and detail. Following the header is a detailed description of the invoice in XML format. The invoice itself is split into a header record (A4) and detail records (A6) as follows:

Main header

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An Id issued by Borgun to the processor using this interface on behalf of the merchant.
MerchantId	<i>Mandatory</i>	An Id in B-Gateway of the merchant requesting services
TerminalId	<i>Optional</i>	A terminal id in B-Gateway for the merchant requesting services.
TransType	<i>Mandatory</i>	Same as in the original transaction
TrAmount	<i>Mandatory</i>	Same as in the original transaction
TrCurrency	<i>Mandatory</i>	Same as in the original transaction
DateAndTime	<i>Mandatory</i>	Same as in the original transaction
PAN	<i>Mandatory</i>	Same as in the original reply
ExpDate	<i>Mandatory</i>	Expiration date – YYMM
RRN	<i>Mandatory</i>	Same as in the original reply
Transaction	<i>Optional</i>	Same as in the original reply
Batch	<i>Optional</i>	Same as in the original reply
AuthCode	<i>Conditional</i>	Same as in the original reply

Invoice header (A4):

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000".
MessageNumber	<i>Mandatory</i>	Message number (sequence number of line within file)
InvoiceNumber	<i>Mandatory</i>	Invoice number
InvoiceDate	<i>Mandatory</i>	Invoice date (yyyymmdd)
Reference	<i>Optional</i>	Reference number
MerchantSSN	<i>Mandatory</i>	Merchant social security number
MerchantVATNumber	<i>Mandatory</i>	Merchant VAT number
Amount1	<i>Mandatory</i>	Total (A6*Amount)
Discount	<i>Optional</i>	Total (A6*DiscountAmount) Discount
AmountWithDiscount	<i>Optional</i>	Total (A6*AmountWithDiscount) Amount with discount
Vgju	<i>Optional</i>	Total (A6*Vgju) (zero filled)
Amount3	<i>Optional</i>	Total (A6*Amount3) (same as AmountWithDiscount)
TotalVAT	<i>Mandatory</i>	Total (A6*VATAmount) VAT
AmountWithVAT	<i>Mandatory</i>	Total (A6*AmountWithVAT) amount with VAT
InvoiceAmount	<i>Mandatory</i>	Total invoice
Rounding	<i>Optional</i>	Round off amount in "aurar"
RikiskaupFee	<i>Optional</i>	Total Rikiskaup fee
GLAccount	<i>Optional</i>	General ledger account number
RecordNumber	<i>Mandatory</i>	Record number within invoice (0001 for header)
NumberOfRecords	<i>Mandatory</i>	Number of records in invoice (A4+number of A6)
SlipNumber	<i>Optional</i>	Slip number
Comments	<i>Optional</i>	Comments
OrderNumber	<i>Optional</i>	Order number
OrderDate	<i>Optional</i>	Order date
DeliveryDate	<i>Optional</i>	Delivery date
DeliveryAddress	<i>Optional</i>	Delivery address

Invoice detail (A6):

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000".
MessageNumber	<i>Mandatory</i>	Message number (sequence number of line within file)
ProductCode	<i>Mandatory</i>	Product code
ItemDescription	<i>Mandatory</i>	Item description
UNSPSCcode	<i>Optional</i>	UNSPSC code
UNSPSCdescription	<i>Optional</i>	UNSPSC description
CommodityCode	<i>Optional</i>	Commodity code
ItemQuantity	<i>Mandatory</i>	Item quantity
UnitOfMeasurement	<i>Optional</i>	Item unit of measure (ltr, km, kg....)
GrossValue	<i>Mandatory</i>	Unit price without VAT
Amount	<i>Mandatory</i>	Amount (ItemQuantity * GrossValue)
DiscountPercent	<i>Optional</i>	Discount percent (%)
DiscountAmount	<i>Optional</i>	Discount amount
AmountWithDiscount	<i>Optional</i>	Amount with discount (Amount – DiscountAmount)
Vgjp	<i>Optional</i>	"00000"
Vgju	<i>Optional</i>	"000000000000"
Amount3	<i>Optional</i>	AmountWithDiscount
VATPercent	<i>Mandatory</i>	VAT %
VATAmount	<i>Mandatory</i>	VAT amount
AmountWithVAT	<i>Mandatory</i>	Amount with VAT (Amount3 + VATAmount)
RikiskaupFeePercent	<i>Optional</i>	Rikiskaup fee (%)
RikiskaupFee	<i>Optional</i>	Rikiskaup fee amount
GLAccount	<i>Optional</i>	General ledger account number
RecordNumber	<i>Mandatory</i>	Record number within invoice (0002-xxxx)
NumberOfRecords	<i>Mandatory</i>	Number of records in invoice (A4+number of A6)
SlipNumber	<i>Optional</i>	Slip number
Ref1	<i>Optional</i>	Reference 1
Ref2	<i>Optional</i>	Reference 2
Ref3	<i>Optional</i>	Reference 3
Ref4	<i>Optional</i>	Reference 4

Table SDD.4 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Same value as in the request message
Processor	<i>Mandatory</i>	Same value as in the request message
MerchantId	<i>Mandatory</i>	Same value as in the request message
TerminalId	<i>Mandatory</i>	Same value as in the request message
TransType	<i>Mandatory</i>	Same value as in the request message
TrAmount	<i>Mandatory</i>	Same value as in the request message
TrCurrency	<i>Mandatory</i>	Same value as in the request message
DateAndTime	<i>Mandatory</i>	Same value as in the request message
PAN	<i>Mandatory</i>	Same value as in the request message
ExpDate	<i>Mandatory</i>	Same value as in the request message
RRN	<i>Mandatory</i>	Same value as in the request message
Transaction	<i>Mandatory</i>	Same value as in the request message
Batch	<i>Mandatory</i>	Same value as in the request message
CardAcclD	<i>Mandatory</i>	Code Identifying the card acceptor. This Id is issued by the acquirer
CardAccTerminal	<i>Mandatory</i>	Code Identifying the terminal the card acceptor is using
AuthCode	<i>Mandatory</i>	Same value as in the request message
ActionCode	<i>Mandatory</i>	Reason code for response

Sending detail data – XML examples

SDDXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<DetailData>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <TransType>1</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401123456783234</PAN>
  <ExpDate>0506</ExpDate>
  <RRN>404</RRN>
  <Batch>3</Batch>
  <A4>
    <Version>1000</Version>
    <MessageNumber>2</MessageNumber>
    <InvoiceNumber>071897</InvoiceNumber>
    <InvoiceDate>20060118</InvoiceDate>
    <MerchantSSN>5415986459</MerchantSSN>
    <MerchantVATNumber>079318</MerchantVATNumber>
    <Amount1>100000</Amount1>
    <TotalVAT>24500</TotalVAT>
    <AmountWithVAT>124500</AmountWithVAT>
    <InvoiceAmount>125000</InvoiceAmount>
    <RecordNumber>1</RecordNumber>
    <NumberOfRecords>1</NumberOfRecords>
  <A6>
    <Version>1000</Version>
    <MessageNumber>2</MessageNumber>
    <ProductCode>054</ProductCode>
    <ItemDescription>Chocolate</ItemDescription>
    <ItemQuantity>5</ItemQuantity>
    <GrossValue>20000</GrossValue>
    <Amount>1000</Amount>
    <VATPercent>0,245</VATPercent>
    <VATAmount>24500</VATAmount>
    <AmountWithVAT>124500</AmountWithVAT>
    <RecordNumber>1</RecordNumber>
    <NumberOfRecords>1</NumberOfRecords>
  </A6>
</A4>
</DetailData>
```

Sending passenger itinerary

This addition to getAuthorization is used to supply passenger itinerary data after the financial transaction has been processed. It currently only supports itineraries with four legs or less.

Sending passenger itinerary – XML explanations

Tables SPI.1, SPI.2 – Field explanations for a sent/received XML

The input is split into a header and detail. Following the header is a detailed description of the invoice in XML format. The invoice itself is split into a header record (A4) and detail records (A6) as follows:

Table SPI.1 – Passenger Itinerary information field explanations for a sent XML

Field	Type	Length	Explanation
Version	<i>Mandatory</i>		Indicates the version of these specifications the client is using. Current version is “1000”
TrDate	<i>Mandatory</i>		Same as in the original transaction
TrTime	<i>Mandatory</i>		Same as in the original transaction
PassengerName	<i>Mandatory</i>	1..29 AN	Name of passenger to whom the ticket was issued
FlightNumber_1	<i>Mandatory</i>	3 AN	Flight number in leg 1
TravelDate_1	<i>Mandatory</i>	6	Traveldate of leg 1, format is YYMMDD.
DepartureAirport_1	<i>Mandatory</i>	3 AN	Departure airport, airport code can be obtained from OAG or equivalent.
CarrierCode_1	<i>Mandatory</i>	2 AN	Carrier code is a standard abbreviation for the airline carrier. Can be obtained from IATA, or an equivalent. Length
ServiceClassCode_1	<i>Optional</i>	3 N	Service type, such as <i>first class</i> , etc. Can be obtained from OAG or equivalent.
DestinationAirport_1	<i>Optional</i>	3 AV	Destination airport, airport code can be obtained from OAG or equivalent.
FareBaseCode_1	<i>Optional</i>	1..15 AN	Fare basis code contains a code that carriers assign to particular ticket type such as business class or discounted/refundable.
FlightNumber_2	<i>Optional</i>		Same as in leg 1
TravelDate_2	<i>Optional</i>		Same as in leg 1
DepartureAirport_2	<i>Optional</i>		Same as in leg 1
CarrierCode_2	<i>Optional</i>		Same as in leg 1
ServiceClassCode_2	<i>Optional</i>		Same as in leg 1
DestinationAirport_2	<i>Optional</i>		Same as in leg 1
FareBaseCode_2	<i>Optional</i>		Same as in leg 1
FlightNumber_3	<i>Optional</i>		Same as in leg 1
TravelDate_3	<i>Optional</i>		Same as in leg 1
DepartureAirport_3	<i>Optional</i>		Same as in leg 1
CarrierCode_3	<i>Optional</i>		Same as in leg 1
ServiceClassCode_3	<i>Optional</i>		Same as in leg 1
DestinationAirport_3	<i>Optional</i>		Same as in leg 1
FareBaseCode_3	<i>Optional</i>		Same as in leg 1
FlightNumber_4	<i>Optional</i>		Same as in leg 1
TravelDate_4	<i>Optional</i>		Same as in leg 1
DepartureAirport_4	<i>Optional</i>		Same as in leg 1

CarrierCode_4	<i>Optional</i>		Same as in leg 1
ServiceClassCode_4	<i>Optional</i>		Same as in leg 1
DestinationAirport_4	<i>Optional</i>		Same as in leg 1
FareBaseCode_4	<i>Optional</i>		Same as in leg 1
TravelAgencyCode	<i>Mandatory</i>	8	Code of the travel agency issuing the ticket
TravelAgencyName	<i>Mandatory</i>	25	Name of travel agency issuing the ticket
TicketNumber	<i>Mandatory</i>	15	Number of the ticket

Sending detail data – XML examples

SDDXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<getAuthorization xmlns="http://tempuri.org/getAuthorization.xsd">
  <Version>1000</Version>
  <Processor>3</Processor>
  <MerchantID>3</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>1</TransType>
  <TrAmount>10000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>120325120000</DateAndTime>
  <PAN>5401710000001234</PAN>
  <ExpDate>1412</ExpDate>
  <CVC2>111</CVC2>
  <RRN>TESTAIR00111</RRN>
  <PassengerItineraryData>
    <A1>
      <Version>10</Version>
      <MessageNumber>1111111</MessageNumber>
      <TrDate>20120222</TrDate>
      <TrTime>151515</TrTime>
      <PassengerName>Jane Doe</PassengerName>
      <FlightNumber_1>111111</FlightNumber_1>
      <TravelDate_1>20120111</TravelDate_1>
      <DepartureAirport_1>KEF</DepartureAirport_1>
      <CarrierCode_1>CC</CarrierCode_1>
      <ServiceClassCode_1>123</ServiceClassCode_1>
      <DestinationAirport_1>JFK</DestinationAirport_1>
      <FareBaseCode_1>111111</FareBaseCode_1>
      <FlightNumber_2>222222</FlightNumber_2>
      <TravelDate_2>20120222</TravelDate_2>
      <DepartureAirport_2>KEF</DepartureAirport_2>
      <CarrierCode_2>CC</CarrierCode_2>
      <ServiceClassCode_2>123</ServiceClassCode_2>
      <DestinationAirport_2>JFK</DestinationAirport_2>
      <FareBaseCode_2>222222</FareBaseCode_2>
      <FlightNumber_3>333333</FlightNumber_3>
      <TravelDate_3>20120330</TravelDate_3>
      <DepartureAirport_3>KEF</DepartureAirport_3>
      <CarrierCode_3>CC</CarrierCode_3>
      <ServiceClassCode_3>123</ServiceClassCode_3>
      <DestinationAirport_3>JFK</DestinationAirport_3>
    </A1>
  </PassengerItineraryData>
</getAuthorization>
```

```

<FareBaseCode_3>333333</FareBaseCode_3>
<FlightNumber_4>444444</FlightNumber_4>
<TravelDate_4>20120404</TravelDate_4>
<DepartureAirport_4>KEF</DepartureAirport_4>
<CarrierCode_4>CC</CarrierCode_4>
<ServiceClassCode_4>123</ServiceClassCode_4>
<DestinationAirport_4>JFK</DestinationAirport_4>
<FareBaseCode_4>444444</FareBaseCode_4>
<TravelAgencyCode>A7654321</TravelAgencyCode>
<TravelAgencyName>All travel</TravelAgencyName>
<TicketNumber>900.123.222</TicketNumber>
</A1>
</PassengerItineraryData>
</getAuthorization>

```

Appendix - Action Codes

Action Code	Explanation	ISO Reply
100-199	Show reason for transaction dismissal by issuer, card pick-up not required.	
000	Accepted	00
100	Do not honor (Not accepted)	05
101	Expired card	54
102	Suspected card forgery (fraud)	
103	Merchant call acquirer	01, 66, 08
104	Restricted card	62
106	Allowed PIN retries exceeded	75, 76
109	Merchant not identified	03
110	Invalid amount	13, 64
111	Invalid card number	14
112	PIN required	
116	Not sufficient funds	51
117	Invalid PIN	55
118	Unknown card	
119	Transaction not allowed to cardholder	57
120	Transaction not allowed to terminal	12, 58
121	Exceeds limits to withdrawal	61, 65
125	Card not valid	
126	False PIN block	
129	Suspected fraud	
130	Invalid Track2	
131	Invalid Expiration Date	
161	DCC transaction not allowed to cardholder	
162	DCC cardholder currency not supported	
163	DCC exceeds time limit for withdrawal	
164	DCC transaction not allowed to terminal	
165	DCC not allowed to merchant	
166	DCC unknown error	
200-299	Show reason for transaction dismissal by issuer, card pick-up required.	
200	No not honor	04
201	Card not valid	33
202	Suspected card forgery (fraud)	

203	<i>Merchant contact acquirer</i>	37
204	<i>Limited card</i>	36
205	<i>Merchant contact police</i>	
206	<i>Allowed PIN-retries exceeded</i>	38
207	<i>Special occasion</i>	
208	<i>Lost card</i>	41
209	<i>Stolen card</i>	43
210	<i>Suspected fraud</i>	
902	<i>False transaction</i>	89, 06
904	<i>Form error</i>	30
907	<i>Issuer not responding</i>	
908	<i>Message can not be routed</i>	15, 80, 81, 92
909	<i>System error</i>	83, 96
910	<i>Issuer did not respond</i>	68
911	<i>Issuer timed out</i>	82
912	<i>Issuer not reachable</i>	91
913	<i>Double transaction transmission</i>	94
914	<i>Original transaction can not be traced</i>	
916	<i>Merchant not found</i>	
950	<i>No financial record found for detail data</i>	
951	<i>Batch already closed</i>	
952	<i>Invalid batch number</i>	
953	<i>Host timeout</i>	
954	<i>Batch not closed</i>	
955	<i>Merchant not active</i>	
956	<i>Transaction number not unique</i>	

FAQ

Frequently Asked Questions

1. Is it possible to perform partial refunds (TransType = 3)? Can this happen multiple times until reaching the total amount authorized?

This is supported, for example

1. Perform a normal purchase transaction (TransType = 1) with amount 10.
2. Then, perform a refund transaction (TransType = 5) with amount 5.
3. Finally, perform another refund transaction for the remaining 5.

The sum of the refunds must be equal or lower than the original sale amount.

2. Is it possible to send normal purchase transactions (TransType = 1) with less amount than the one specified in the original pre-authorization (TransType = 5) so that the final authorized amount is the one sent in the normal purchase?

This is supported, for example

1. Perform a pre-authorize transaction (TransType = 5) with amount 10.
2. Capture the previous pre-authorization performing a normal purchase transaction (TransType = 1) with amount 5.

Note that the cardholder will have the withdrawal limit corrected 1-2 days later.

3. Has RRN any kind of uniqueness restriction?

RRN has must be size 12 alphanumeric, the last 6 letters cannot be 000000. For reference sake it is recommended to be a sequence with fixed first part, i.e. ACME12345678

4. Are batches automatically closed at the same time every day? In that case, could you tell us the exact time and time zone when this happens? Can this time be configured or is it fixed?

Default is to have the batch close time fixed, then it will close at midnight GMT-0 (no daylight changes).

It can be controlled by calling party by using the newBatch() command, at execution the old batch is closed and a new one is opened. This is the preferred way to use for custom closure.

5. Should Refund or Partial Reversal be used?

Refund should be preferred in general, it informs the issuer immediately that the authorization amount will be changed and the credit amount is on the cardholder account in 24-48hours.

Partial Reversal is more suited to specific business cases where there is a high likelihood of an immediate change in the sale amount. An example of this is a vending machine that authorizes for the maximum amount at the beginning then has the sale amount as the actual item buying value.

6. What is the date format

The gateway uses either yyyyymmddhhmmss or yymmddhhmmss depending on functions.

The format is:

yyyy: year in numeric form with four letters, f.ex. 2012 (or 12 if yy)

mm: month in numeric format (f.ex. 12 as december)

dd: date in numeric format (f.ex. 31)

hh: hour in 24 hour numeric format (f.ex. 18)

mm: minutes (f.ex. 18)

ss: seconds (f.ex. 59)

example

20121231184059

7. When are batches closed and financial records sent?

Batches are closed at GMT+0 with no daylight savings. All dates at Borgun are running at GMT+0. Financial records are sent during the night. Merchants can control the batch close time by using `newBatch()`, the use cases for that are rare and tied to specific merchant segments.

8. 3D-Secure, when is XID needed and where is VISA ECI

MasterCard uses UCAF and SecurityLevelIndicator, there is no XID in MasterCard SecureCode.

VISA uses CAVV, XID and SecurityLevelIndicator. SecurityLevelIndicator is VISA ECI, the insert value is converted to the equivalent ECI value.

While CAVV and UCAF have the same function, and SecurityLevel and ECI also have the same function, then the XID is simply a merchant reference number and only used by VISA.

9. How are credits differentiated/displayed in the settlement response

Credits and deductions are represented with the minus sign in settlement reconciliation data.

10. What is the preferred timeout?

Borgun gateway waits for 35 seconds before cancelling requests. A suitable timeout is 40 seconds.

11. How often can refund be sent for the original transaction?

The cumulative amount of refunds cannot exceed the amount on the original transaction. The refunds can be as many as needed until the sum is equal to the original amount.

12. What configuration items are used?

As merchants are processed through the business channels they are registered and setup according to business usage.

The Business channel will send back

- MerchantId: used in payment gateway to send in authorization request
- MID: used in payment gateway when fetching settlement/fraud/dispute, i.e. reconciliation data
- Terminal information, usually some combination of the following terminal types

Terminal 1 is ecommerce

Terminal 2 is ecommerce with 3D-Secure

Terminal 3 is recurrent

Terminal 4 is MOTO

It simplifies configuration setup if terminal 1 is always ecommerce and 4 is always MOTO etc., but if needed then they can be put in any order.

Not all merchants have all terminals types and would then be sent for example the following

Terminal 1 is ecommerce

Terminal 4 is MOTO

During initial setup a processor number is created that will follow all requests from the merchant, processor, ISO.

13. Which ActionCodes constitutes errors and which success?

Transactions will only be marked as approved if the returned code from issuer is 000.

There are four ActionCodes that Borgun does returns, the rest are exclusively from issuers

904: Malformed XML or incorrect data in insert XML

909: System error or network connection to schema is lost

953: Host timeout (can also come from network, but then sporadically)

The difference between 909 and 953 is minimal, if 953 then we will send the request and wait for response, if 909 then we know that the line is down and there is no reason to send the request.

If there are problems, then they start with 953 and then change to 909 until the we problem is fixed.

Note that 904, 909, 953 can also come from issuers or the card schema network.

904 should always be looked into.

14. How to find information on authorization if no response is received?

At midnight all open batches are closed, financial records are sent to the card schema and transaction status on financial transactions is set to value 14. `getTransactionList()` can be called on a regular basis to check on status value and use to reconcile authorization requests with unknown status.

15. Do the xml elements in a request xml string parameter need to be in a particular order?

No, the xml elements can be in any order.

16. Common pitfalls?

- Refunds and `cancelAuthorization` need the exact `DateAndTime` + `RRN` of the original transaction that is being referred.
- `ExpireDate` in all authorization requests is of the form `YYMM`

- Using cancelAuthorization or Partial Reversal against an original transaction that is in closed batch. After batch is closed the transaction cannot be changed but Refund can be used.
- For historical reasons then the merchantid tag is <MerchantId> in some functions and <MerchantID> in others.
- Sending in authorization request without TerminalId. The request will go through processing and return 909, the reason being that it is allowed to send in requests without terminal if one terminal has been marked as Default. That function is not in common use.