Version 1.4

User Guide

Merchant Swish Simulator

Date: 2019-02-18

swish®

Table of content

1 Revision history

Date	Version	Name	Description
2015-11-05	0.9.8	AT	First release to publish
2015-12-10	0.9.8.1	CS	Renamed document to Guide Testverktyg 3.3 Added information about port. 4.2.1 Changed example callback url. 4.2.2 Changed example callback url. 4.3 Changed example callback url. 4.5 Deleted error codes AC05, AC06, AC07, AC15, AM04, AM14, AM21, and DS0K. Added error code RF07. 5.1 Changed example callback url. 5.2 Changed example callback url. 6.2 Changed example callback url. 6.3 Changed example callback url. 6.5 Deleted error codes AC05, AC06, AC07, AC15, AM04, AM14, AM21, and DS0K. Added error code RF07. Deleted note related to error code RF04. Added error code RF07. 7.1 Changed example callback url.
2016-01-18	1.0	PJ	Created version 1.0
2018-08-15	1.1	Magnus Lageson	Updated end-points
2018-10-24	1.2	Magnus Lageson	Updated Curl calls
2018-10-26	1.3	Mats Bergström	Document title: Renamed to ‘User Guide’. Ch. 2: Renamed from ‘Background’ to ‘Introduction’ and updated. Ch. 3.2: Added

			information about Technical Supplier certificate. Ch. 4.4; 6,4: Added information about MSS limitation for HTTP 403. Ch: 4.5; 6.5: Elaborated on how error codes can be used to simulate errors and added new codes applicable for Technical Supplier API user. Ch 7: Added information about MSS cache.
2019-02-18	1.4	Mats Bergström	Document updated with information about new optional parameters `payerSSN` and `ageLimit` for the Payment Request API. In addition all `curl` examples has been completely updated and verified. A new chapter regarding `curl` usage has been added and a number of old chapters has been renamed to better reflect their content as well updated with more information on how to use them.

2 Introduction

This document is intended for Merchants and Technical Suppliers (TS) who wish to verify their usage of the *Swish* Payment and Refund Request API.

The Merchant Swish Simulator (MSS) is a test tool to offer a way for merchants and technical suppliers to verify the format and content of API calls to Swish *without* any backward integration to other system components. An API call will return an error message or a correctly formatted response message depending on what data is put in the provided request data object. This allows for the API user to validate both positive and negative (error) scenarios.

3 API usage

3.1 Endpoints simulated by MSS

MSS provides two endpoints (URL) for simulation of *Swish* payment/refund requests:

`https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/`

`https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/`

These are REST endpoints and the data to provide in the request data object (body) should be in JSON format.

Important to note is that contrary to the real Swish system there is no connection between these two endpoints in MSS. They are completely independent of each other meaning that data created/retrieved through the paymentrequests endpoint are not available through the refunds endpoint.

3.2 TLS certificates

In order to communicate with the *MSS server* the API client should use one of the provided client TLS certificate files:

- Swish_Merchant_TestCertificate_1231181189.p12
- Swish_TechnicalSupplier_TestCertificate_9871065216.p12

These files contains a *swish registered* test certificate together with the complete chain of trust and the private key. The number specified in the file name is the swish number (`payeeAlias`) for which the certificate was created. The password for the private key is *swish* for both certificates.

Alternatively, the Merchant/Technical Supplier could use its own *swish registered* certificate, non-registered certificates will not work but result in an error response (or no response at all).

Normally, when a client sends a create payment request it must set the `payeeAlias` (swish number) data field to the number for which the certificate was created (e.g. 1231181189) or the request would be rejected. Note that MSS will *not reject* any payment requests due to mismatch between `payeeAlias` and used certificate, any `payeeAlias` value can be used as long as it is a properly formatted swish number and the used certificate is a valid one.

The Swish Root CA certificate is available in the files mentioned above but also in a separate file "Swish_TLS_RootCA.pem".

3.3 TLS for the callback endpoint

The client *callback endpoint* has to use HTTPS on port 443 and it is highly recommended to use IP filtering as well. For the callback MSS will be acting client and the merchant/technical supplier is acting server. Swish will validate the callback server TLS certificate against a list of commonly recognized Certificate Authorities (CA).

3.4 Call endpoints using `curl`

This document includes a number of examples on how one can manually call the API using `curl`, a command line tool for transferring data with URLs. Both positive and negative examples are provided.

To minimize the risk of running into `curl` tool related problems it is recommended to run the examples using a a fairly new version of `curl` built with *openSSL* in favour of *GnuTLS.* (the default version for Ubuntu 16.04 is 7.47.0 built with GnuTLS and will most likely not work but result in an error message: curl: (35) error reading X.509 potentially-encrypted key file: Base64 decoding error.)

For details on how to use `curl`, please refer to the man page.

All examples has been verified to work as expected when executed on a GNU/Linux machine with:

- Linux Kernel version: #61~16.04.1-Ubuntu SMP Wed Jun 14 11:58:22 UTC 2017
- curl version: curl 7.61.1 (x86_64-pc-linux-gnu) libcurl/7.61.1 OpenSSL/1.0.2g zlib/1.2.8

In the provided examples the used `curl` commands are built up like this:

```
$ curl -s -S -i --cert <path-to-certificate-file>:<password> --cert-type
p12 --cacert <path-to-rootCA-pem-file> --tlsv1.1 --header "Content-Type:
application/json" <endpoint-url> --data '<json-formatted-data>'
```

All examples are executed from the directory in the file system where the certificate files are located.

Note on copy-paste of example `curl` commands: Some PDF viewers insert newline characters on copy. Thus, if you want to copy an example command it might be necessary to first paste the copy into an editor and remove any newline characters before pasting into a shell.

4 Create payment request (POST)

Merchants and Technical Suppliers can send create payment requests for both E-Commerce and M-Commerce to MSS.

In order to send create payment request the request object needs to be a POST to URL:

`https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/`

Once MSS receives a “create payment request” call, there are two answers that will be returned from MSS (unless error situation). The first answer is synchronous, the second one is asynchronous.

Please note the following:

1. MSS directly returns a “Payment request created” response to the caller.
   In case of M- Commerce the response will contain a *PaymentRequestToken* in the header in addition to the *Location* property. This token is unique for each payment request.
2. MSS sends a “Payment result” to the provided callback URL after some delay, normally four seconds. This delay is not configurable by the client.
   For the M-Commerce case the MSS server will set the `payerAlias` property to a fake value of “46464646464” in the response.
3. If the client simulates an error situation by providing an error code in the `message` property the error response could come immediately or in the callback depending on used error code.

4.1 Payment request data object

The Payment Request Object is used in all payment request operations and the provided data object (body) should be in JSON format. The table shows all possibly properties for a payment request, the properties marked as mandatory (M) must be present or the call will be rejected.

Property	Type	Mandatory/ Optional	Description
payeePaymentReference	string	O	Payment reference supplied by theMerchant. This is not used by Swish but is included in responses back to the client. This reference could for example be an order id or similar. If set the value must not exceed 35 characters and only the following characters are allowed: [a-ö, A-Ö, 0-9, -]

callbackUrl	string	M	URL that Swish will use to notify caller about the result of the payment request. The URL has to use HTTPS.
payerAlias	string	O	The registered Cell phone number of the person that makes the payment. It can only contain numbers and has to be at least 8 and at most 15 digits. It also needs to match the following format in order to be found in Swish: country code + cell phone number (without leading zero). E.g.: 46712345678 If set, request is handled as E-Commerce payment. If not set, request is handled as M-Commerce payment.
payerSSN	String	O	The social security number of the individual making the payment, should match the registered value for *payerAlias* or the payment will not be accepted. The value should be a proper Swedish social security number (personnummer or sammordningsnummer). Note: Since MSS is a stand-alone test system it can not verify if *payerSSN* match registered value for *payerAlias*. It is possible to simulate an ‘ssn not matching’ error, see *message* property below.
ageLimit	String	O	Minimum age (in years) that the individual connected to the *payerAlias* has to be in order for the payment to be accepted. Value has to be in the range of 1 to 99. Note: Since MSS is a stand-alone test system it can not verify the *payerAlias* age against the *ageLimit* value. It is possible to simulate an ‘age to low’ error, see *message* property below.
payeeAlias	string	M	The Swish number of the payee. It needs to match with Merchant Swish number.

amount	string	M	The amount of money to pay. The amount cannot be less than 1 SEK and not more than 999999999999.99 SEK. Valid value has to be all digits or with 2 digit decimal separated with a period.
currency	string	M	The currency to use. Currently the only supported value is SEK.
message	string	O	Merchant supplied message about the payment/order. Max 50 characters. Allowed characters are the letters a-ö, A-Ö, the numbers 0-9 and any of the special characters :;.,?!()-”. For MSS, an *error code* as defined in section 4.4 can be set in this *message* property in order to simulate an error situation.

4.2 Examples

4.2.1 Successful payment request, E-Commerce

A create payment request *with* the payer mobile telephone (`payerAlias`) number set is for E-Commerce.

```
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests --data
'{ "payeePaymentReference" : "0123456789", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "payerAlias" : "4671234768",
"payeeAlias" : "1231181189", "amount" : "100", "currency" : "SEK",
"message" : "Kingston USB Flash Drive 8 GB" }'

HTTP/1.1 201 Created
Location:
https://mss.cpc.getswish.net/swish-cpcapi/v1/paymentrequests/
AB23D7406ECE4542A80152D909EF9F6B

After some delay the provided callbackURL is called by MSS with the result
of the payment request.
```

4.2.2 Successful payment request, M-Commerce

A create payment request *without* the payer mobile telephone number (`payerAlias`) is for M-Commerce.

```
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests --data
'{ "payeePaymentReference" : "0123456789", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "payeeAlias" : "1231181189",
"amount" : "100", "currency" : "SEK", "message" : "Kingston USB Flash Drive
8 GB" }'

HTTP/1.1 201
Location: https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/
98BF074EE6CA42F7BCBE175182D59659
Server: nginx/1.12.1
Connection: keep-alive
PaymentRequestToken: 00132ec0dda74b12acc142fa355181fc
Content-Length: 0
Date: Tue, 12 Feb 2019 14:22:21 GMT

After some delay the provided callbackURL is called by MSS with the result
of the payment request.
```

4.2.3 Error, invalid parameter

A create payment request with an invalid `payeeAlias` property.

```
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests --data
'{ "payeePaymentReference" : "0123456789", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "payerAlias" : "4671234768",
"payeeAlias" : "9991181189", "amount" : "100", "currency" : "SEK",
"message" : "Kingston USB Flash Drive 8 GB" }'

HTTP/1.1 403
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 Feb 2019 14:51:24 GMT

[{"errorCode":"PA01","errorMessage":"Parameter is not
correct.","additionalInformation":""}]
```

4.2.4 Simulating an error using message property

A create payment request with error code BE18 in `message` property, simulating an error situation. For a list of error codes refer to section 4.4.

```
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests --data
'{ "payeePaymentReference" : "0123456789", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "payerAlias" : "4671234768",
"payeeAlias" : "1231181189", "amount" : "100", "currency" : "SEK",
"message" : "BE18" }'

HTTP/1.1 422 Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked Date: Tue, 12 Feb 2019 14:32:09 GMT

[{"errorCode":"BE18","errorMessage":"Payer alias is
invalid","additionalInformation":null}]
```

4.3 HTTP status codes

The following HTTP status codes could be received in a create payment response message:

HTTP status codes	Returned scenarios
201 Created	Returned when Payment request was successfully created. Will return a `Location` header. For the M-Commerce case the response will also contain a `PaymentRequestToken` header.
400 Bad Request	Returned when the Create Payment Request operation was malformed.
401 Unauthorized	Returned when there are authentication problems with the certificate. Or the Swish number in the certificate is not enrolled. Will return nothing else.
403 Forbidden	Returned when the `payeeAlias` in the payment request object is not the same as Merchants Swish number in the provided certificate or the value is not a valid swish number. Note: Since MSS is a test system, any mismatch between the swish number in provided certificate and the request data object are ignored by MSS, it is only logged as a mismatch in the server.
415 Unsupported Media Type	Returned when Content-Type header is not "application/json". Will return nothing else.
422 Unprocessable Entity	Returned when there are validation errors.
500 Internal Server Error	Returned if there was some unknown/unforeseen error that occurred on the server, this should normally not happen. Will return nothing else.

4.4 Error Simulation codes

The client can trigger (simulate) an error situation by setting the `message` property of the *create payment request* data object to an appropriate value as listed in the following sections.

Depending on error code and type of payment (E-commerce/M-commerce) the simulated error will be triggered immediately in the first ‘Payment request created’ step or in the second delayed ‘Payment result’ step (callback).

4.4.1 Simulate failure for the ‘*Payment request created’* step

Simulate failure for the ‘*Payment request created’* step (first response):

Setting the *message* property to one of the following error codes will result in an error response message with HTTP status code *422*. The included response object will contain the error code in its `errorCode` property and a description in the `errorMessage` property.

Error codes	Description
FF08	PayeePaymentReference is invalid
RP03	Callback URL is missing or does not use Https
BE18	Payer alias is invalid
RP01	Payee alias is missing or empty
PA02	Amount value is missing or not a valid number
AM06	Amount value is too low
AM02	Amount value is too large
AM03	Invalid or missing Currency
RP02	Wrong formatted message
RP06	Another active PaymentRequest already exists for this payerAlias. (Only applicable for E-Commerce.)
ACMT03	Payer not Enrolled
ACMT01	Counterpart is not activated
ACMT07	Payee not Enrolled
UNKW	Technical supplier is not active (only applicable for Technical Supplier API user).
VR01	Does not meet age limit. Note: Only E-Commerce case, for M-Commerce case the simulated error will trigger in the delayed ‘Payment result callback’ (second response).
VR02	SSN does not match enroled customer. Note: Only E-Commerce case, for M-Commerce case the simulated error will trigger in the delayed ‘Payment result callback’ (second response).

Setting the *message* property to one of the following error codes will result in an error response message with HTTP status code *403*. The included response object will contain the error code in its `errorCode` property, no detailed information is provided.

Error codes	Description
PA01	The technical supplier is not connected to the merchant (only applicable for Technical Supplier API user).

4.4.2 Simulate failure for the ‘*Payment result’* step

Simulate failure for the ‘*Payment result’* step (second answer, data provided in callback):

Setting the `message` property to one of the following error codes will result in the `status` property of the *callback* data object to be set to `ERROR` and the `errorCode` property is set to the error code simulated.

Note: The simulated error situation is also indicated in any following GET requests sent to MSS, *refer to section 5.2 for an example.*

Error codes	Description
RF07	Transaction declined
BANKIDCL	Payer cancelled BankId signing
FF10	Bank system processing error
TM01	Swish timed out before the payment was started
DS24	Swish timed out waiting for an answer from the banks after payment was started. Note: If this happens Swish has no knowledge of whether the payment was successful or not. The Merchant should inform its consumer about this and recommend them to check with their bank about the status of this payment.
VR01	Does not meet age limit. Note: Only M-Commerce case, for E-Commerce case the simulated error will trigger in the ‘Payment request created’ step (first response).
VR02	SSN does not match enroled customer. Note: Only M-Commerce case, for E-Commerce case the simulated error will trigger in the ‘Payment request created’ step (first response).

5 Retrieve payment result (GET)

The client can retrieve payment result information of an initiated payment request (successful or failed) by sending a GET request to the following URL:

```
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/<id>
```

The complete URL (with the payment request identifier) can be found in the HTTP `Location` header returned in the response message for a previous *create payment request* call.

If the previous *create payment request* call simulated a delayed error the response of the GET operation will have a `status` property with value ERROR and properties `errorCode` and `errorMessage` will be set accordingly. See chapter 4.4.2. for possible errors that can be simulated with a delay.

For complete examples of positive and negative cases please refer to sections 5.1 and 5.2.

Remark: MSS stores the necessary information about each incoming “Payment request” in a cache which automatically expires every 24 hours or when the MSS server is restarted.

5.1 Example, Retrieve payment request result

```
Create payment request:
$ curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"  https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests
--data '{ "payeePaymentReference" : "0123456789", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "payeeAlias" : "1231181189", "amount" :
"100", "currency" : "SEK", "message" : "Kingston USB Flash Drive 8 GB" }'

HTTP/1.1 201
Location:
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/98BF074EE6CA42F7BCB
E175182D59659
Server: nginx/1.12.1
Connection: keep-alive
PaymentRequestToken: 00132ec0dda74b12acc142fa355181fc
Content-Length: 0
Date: Tue, 12 Feb 2019 14:22:21 GMT

After delay time, Retrieve the payment result:
curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/98BF074EE6CA42F7BCB
E175182D59659

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 Feb 2019 15:33:34 GMT

{"id":"98BF074EE6CA42F7BCBE175182D59659", "payeePaymentReference":"0123456789",
"paymentReference":"6E7DBB1C56CB43E787CBB18F906A585D",
"callbackUrl":"https://myfakehost.se/swishcallback.cfm", "payerAlias":"46464646464",
"payeeAlias":"1231181189", "amount":100.00, "currency":"SEK", "message":"Kingston
USB Flash Drive 8 GB", "status":"PAID", "dateCreated":"2019-02-12T14:22:21.610Z",
"datePaid":"2019-02-12T14:22:25.610Z", "errorCode":null, "errorMessage":null}
```

5.2 Example. Retrieve payment with simulated error

```
Create payment request with a simulated delayed error (M-commerce, VR01):
curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish
--cert-type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header
"Content-Type: application/json" https://mss.cpc.getswish.net/swish-cpcapi/
api/v1/paymentrequests --data '{ "payeePaymentReference" : "0123456789",
"callbackUrl" : "https://myfakehost.se/swishcallback.cfm",  "payeeAlias" :
"1231181189", "amount" : "100", "currency" : "SEK", "ageLimit": "18",
"message" : "VR01" }'

HTTP/1.1 201
Location: https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests/
BC4C3F91A3FD44D58F97D0A11C08EA34
Server: nginx/1.12.1
Connection: keep-alive
PaymentRequestToken: ff4937140f214daaa7c946aaa8cd09f8
Content-Length: 0
Date: Tue, 12 Feb 2019 16:11:46 GMT

After delay time, Retrieve the payment result:
curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish
--cert-type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header
"Content-Type: application/json" https://mss.cpc.getswish.net/swish-cpcapi/
api/v1/paymentrequests/BC4C3F91A3FD44D58F97D0A11C08EA34

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 Feb 2019 16:12:30 GMT
{"id":"BC4C3F91A3FD44D58F97D0A11C08EA34",
"payeePaymentReference":"0123456789",
"paymentReference":"CC3F94D57A54451DA0F2CEDFD89C24FE",
"callbackUrl":"https://myfakehost.se/swishcallback.cfm",
"payerAlias":"46464646464","payeeAlias":"1231181189","amount":100.00,
"currency":"SEK", "message":"VR01", "status":"ERROR", "dateCreated":"2019-
02-12T16:11:46.040Z", "datePaid":null, "errorCode":"VR01",
"errorMessage":"Does not meet age limit"}
```

5.3 HTTP status codes

The following HTTP status codes could be received in a retrieve payment result response:

HTTP status codes	Returned scenarios
200 OK	Returned when Payment request was found. Will return Payment Request Object. The *status* property will indicate if the payment was successful or not.
401 Unauthorized	Returned when there are authentication problems with the certificate. Will return nothing else.
404 Not found	Returned when the Payment request was not found. Will return nothing else.
500 Internal Server Error	Returned if there was some unknown/unforeseen error that occurred on the server, this should normally not happen. Will return nothing else.

6 Create refund request (POST)

Merchants and Technical Suppliers can send create refund request to MSS.

In order to send create refund request the request object needs to be POST to URL:

https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/

When MSS receives a “Refund request” there are three answers that will be returned from MSS (unless error situation). The first answer is synchronous, the second and third responses are asynchronous.

Please note the following:

1. Contrary to the real Swish system there is no connection between the ‘Create payment request’ and ‘Create refund request’ flows in MSS. Thus, MSS can not validate that the values for `originalPaymentRequest`, nor `amount` are valid.
   The user can however simulate errors by setting the message property to appropriate error code. Refer to section 6.5 for possible error codes
2. MSS directly returns a “Refund request created” response to the caller.
3. MSS sends an intermediate “Refund result” (status DEBITED) to the provided callback URL after some delay, normally four seconds. The delay is not configurable by the client.
4. MSS sends a final “Refund result” (status PAID) to the provided callback URL after a second delay, normally four seconds. The delay is not configurable by the client.

6.1 Refund request object

The Refund request object is used in all refund operations and the provided data object (body) should be in JSON format. The table shows all possibly properties for a payment request, the properties marked as mandatory (M) must be present or the call will be rejected.

Property	Type	Mandatory/ Optional	Description
payerPaymentReference	string	O	Payment reference supplied by the Merchant. This is not used by Swish but is included in responses back to the client.
originalPaymentReference	string	M	Payment reference to the original payment that this refund is for.
callbackUrl	string	M	URL that Swish will use to notify caller about the outcome of the refund. The URL has to use HTTPS.

payerAlias	string	M	The Swish number of the Merchant that makes the refund payment.
payeeAlias	string	O	The Cell phone number of the person that receives the refund payment.
amount	string	M	The amount of money to refund. The amount cannot be less than 1 SEK and not more than 999999999999.99 SEK. Moreover, the amount cannot exceed the remaining amount of the original payment that the refund is for. Note that MSS actually do not check if the refund is greater than the remaining amount of the original payment.
currency	string	M	The currency to use. Only supported value currently is SEK.
message	string	O	Merchant supplied message about the refund. Max 50 chars. Allowed characters are the letters a-o¨, A-Ö, the numbers 0-9 and the special characters :;.,?!()”. For MSS, an error code as defined in section6.5 can be set in this message property in order to simulate an error situation.

6.2 Example, Successful refund request

```
Create refund request:
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net:443/swish-cpcapi/api/v1/refunds --data
'{ "payerPaymentReference" : "0123456789", "originalPaymentReference" :
"6D6CD7406ECE4542A80152D909EF9F6B", "callbackUrl" : "https://myfakehost.se/
swishcallback.cfm", "amount" : "100", "currency" : "SEK", "payerAlias" :
"1234567839", "payeeAlias" : "9991234569", "message": "Refund for Kingston
SSD Drive 320 GB"}'

HTTP/1.1 201
Location:
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/D77BE41AF953468CAD
CA21D244724941
Server: nginx/1.12.1
Connection: keep-alive
Content-Length: 0D
Date: Mon, 18 Feb 2019 16:00:05 GMT

After some delay the provided callbackURL is called by MSS twice (unless
érror situation) with the result of the payment request.
```

6.3 Example, Simulating an error using message property

A “Refund request” call with error code ACMT07 in `message` property simulating an error situation.

```
Create refund request:
$ curl -s -S -i --cert
./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-type p12 --
cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net:443/swish-cpcapi/api/v1/refunds --data
'{ "payerPaymentReference" : "0123456789", "originalPaymentReference" :
"6D6CD7406ECE4542A80152D909EF9F6B", "callbackUrl" : "https://myfakehost.se/
swishcallback.cfm", "amount" : "100", "currency" : "SEK", "payerAlias" :
"1234567839", "payeeAlias" : "9991234569", "message": "ACMT07"}'

HTTP/1.1 422
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 18 Feb 2019 16:12:33 GMT

[{"errorCode":"ACMT07","errorMessage":"Payee alias not
enrolled","additionalInformation":null}]
```

6.4 HTTP status codes

The following HTTP status codes could be received in a create refund response:

HTTP status codes	Returned scenarios
201 Created	Returned when Refund was successfully created. Will return a `Location` header.
400 Bad Request	Returned when Create refund POST operation was malformed.
401 Unauthorized	Returned when there are authentication problems with the certificate. Or the Swish number in the certificate is not enrolled. Will return nothing else.
403 Forbidden	Returned when the `payerAlias` in the refund object is not the same as Merchants Swish number in the provided certificate. Note: This is not the case for MSS. Any mismatch between the swish number in provided certificate and the request data object are ignored by MSS, it is only logged as a mismatch in the MSS server. The only case where 403 is returned is if the API user has deliberately simulated an error resulting in 403.
415 Unsupported Media Type	Returned when Content-Type header is not "application/json". Will return nothing else.
422 Unprocessable Entity	Returned when there are validation errors. Will return an Array of Error Objects.
500 Internal Server Error	Returned if there was some unknown/unforeseen error that occurred on the server, this should normally not happen. Will return nothing else.
504 Gateway Timeout	Returned when the Bank validation answers take too long and Swish times out. This rarely happens.

6.5 Error simulation codes

The API user can trigger (simulate) an error situation by setting the `message` property of the *Create refund request* data object to appropriate value as indicated in the following tables.

Simulate failure for the '*Refund response'* step (first response):

Setting the *message* property to one of the following error codes will result in an error response message with HTTP status code *422*. The included response object will contain the error code in its `errorCode` property and the description in the `errorMessage` property.

Error codes	Description
FF08	PayerPaymentReference is invalid
RP03	Callback URL is missing or does not use Https
PA02	Amount value is missing or not a valid number
AM06	Amount value is too low
RF08	Amount value is too large or amount exceeds the amount of the original payment minus any previous refunds. Note: the remaining available amount is put into the additional information field.
AM03	Invalid or missing Currency
RP01	Payer alias is missing or empty
RP02	Invalid Message text
ACMT07	Payee not Enrolled
ACMT01	Counterpart is not activated
RF02	Original Payment not found or original payment is more than than 13 months old
RF03	Payer alias in the refund does not match the payee alias in the original payment.
RF04	Payer organization number does not match original payment payee organization number.
RF06	The Payee SSN (personnummer) in the original payment is not the same as the SSN for the current Payee. Note: Typically this means that the Mobile number has been transferred to another person.
BE18	Invalid contact details error
UNKW	Technical supplier is not active (only applicable for Technical Supplier API user).

Setting the *message* property to one of the following error codes will result in an error response message with HTTP status code *403*. The included response object will contain the error code in its `errorCode` property.

Error codes	Description
PA01	The technical supplier is not connected to the merchant (only applicable for Technical Supplier API user).

Simulate failure for the '*Refund result'* step (second answer, data provided in callback):

Setting the *message* property to one of the following error codes will result in the `status` property of the *callback* data object will be set to `ERROR` and the `errorCode` property is set to the error code simulated.

Error codes	Description
RF07	Transaction declined
BANKIDCL	Payer cancelled BankId signing
FF10	Bank system processing error
DS24	Swish timed out waiting for an answer from the banks after payment was started. Note: If this happens Swish has no knowledge of whether the payment was successful or not. The Merchant should inform its consumer about this and recommend them to check with their bank about the status of this payment.

7 Retrieve refund result (GET)

The client can retrieve refund result information of an initiated refund request (successful or failed) by sending a GET request to the following URL:

https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/<id>

The complete URL (with the refund request identifier) can be found in the HTTP `Location` header returned in the response message for a previous *create refund request* call.

If the previous *create refund request* call simulated a delayed error the response of the GET operation will have a `status` property with value ERROR and properties `errorCode` and `errorMessage` will be set accordingly. See chapter 6.5 for possible errors that can be simulated with a delay.

Remark: MSS stores the necessary information about each incoming “Refund request” in a cache which automatically expires every 24 hours or when the MSS server is restarted.

7.1 Example, Retrieve refund result

```
Create refund request:
$ curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json" https://mss.cpc.getswish.net:443/swish-cpcapi/api/v1/refunds --
data '{ "payerPaymentReference" : "0123456789", "originalPaymentReference" :
"6D6CD7406ECE4542A80152D909EF9F6B", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "amount" : "100", "currency" : "SEK",
"payerAlias" : "1234567839", "payeeAlias" : "9991234569", "message": "Refund for
Kingston SSD Drive 320 GB"}'

HTTP/1.1 201
Location:
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/D77BE41AF953468CADCA21D2447
24941
Server: nginx/1.12.1
Connection: keep-alive
Content-Length: 0D
Date: Mon, 18 Feb 2019 16:00:05 GMT

After delay time, Retrieve refund result:
$ curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/D77BE41AF953468CADCA21D2447
24941

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 18 Feb 2019 17:34:43 GMT
{"id":"D77BE41AF953468CADCA21D244724941",
"paymentReference":"20CC72A882A64488ABC0878C623043EF",
"payerPaymentReference":"0123456789",
"originalPaymentReference":"6D6CD7406ECE4542A80152D909EF9F6B",
"callbackUrl":"https://myfakehost.se/swishcallback.cfm", "payerAlias":"1234567839",
"payeeAlias":"9991234569", "amount":100.00, "currency":"SEK", "message":"Refund for
Kingston SSD Drive 320 GB", "status":"PAID", "dateCreated":"2019-02-
18T16:00:05.484Z", "datePaid":"2019-02-18T16:00:13.605Z", "errorMessage":null,
"additionalInformation":null, "errorCode":null}
```

7.2 Example, Retrieve refund with simulated error

```
Create refund request with delayed simulated error:
$ curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json" https://mss.cpc.getswish.net:443/swish-cpcapi/api/v1/refunds --
data '{ "payerPaymentReference" : "0123456789", "originalPaymentReference" :
"6D6CD7406ECE4542A80152D909EF9F6B", "callbackUrl" :
"https://myfakehost.se/swishcallback.cfm", "amount" : "100", "currency" : "SEK",
"payerAlias" : "1234567839", "payeeAlias" : "9991234569", "message": "DS24"}'

HTTP/1.1 201
Location:
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/DB68078911964A73A60D0D0507F
ACECF
Server: nginx/1.12.1
Connection: keep-alive
Content-Length: 0
Date: Mon, 18 Feb 2019 17:48:46 GMT

After delay time, Retrieve refund result:
$ curl -s -S -i --cert ./Swish_Merchant_TestCertificate_1231181189.p12:swish --cert-
type p12 --cacert ./Swish_TLS_RootCA.pem --tlsv1.1 --header "Content-Type:
application/json"
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds/DB68078911964A73A60D0D0507F
ACECF

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 18 Feb 2019 17:50:26 GMT

{"id":"DB68078911964A73A60D0D0507FACECF","paymentReference":null,"payerPaymentRefere
nce":"0123456789","originalPaymentReference":"6D6CD7406ECE4542A80152D909EF9F6B","cal
lbackUrl":"https://myfakehost.se/
swishcallback.cfm","payerAlias":"1234567839","payeeAlias":"9991234569","amount":100.
00,"currency":"SEK","message":"DS24","status":"ERROR","dateCreated":"2019-02-
18T17:48:46.264Z","datePaid":null,"errorMessage":"Swish timed out waiting for an
answer from the banks after payment was
started","additionalInformation":null,"errorCode":"DS24"
```

7.3 HTTP status codes

The following HTTP status codes could be received in a retrieve refund result response:

HTTP status codes	Returned scenarios
200 OK	Returned when refund was found. Will return Refund Object..
401 Unauthorized	Returned when there are authentication problems with the certificate. Will return nothing else.
404 Not found	Returned when no refund was found or it was not created by the Merchant. Will return nothing else. Note that for MSS there is no check that the refund was created by the merchant that tries to retrieve the refund result. As long as the refund exist the result will be returned to the calling client.
500 Internal Server Error	Returned if there was some unknown/unforeseen error that occurred on the server. Will return nothing else.

8 Tips on how to trigger an error

In addition to using the `message` property for simulating errors in the synchronous and asynchronous responses other methods can be used. These are described below:

- HTTP communication
  - Provide an incorrect address e.g. i.e. remove “s” in paymentrequests – HTTP 404 Not Found
  - Remove client certificate – “Received fatal alert: handshake_failure”
- Payment Request
  - "payeePaymentReference"
    - Provide too long – “FF08","errorMessage":"Payment Reference is invalid"
    - Provide NULL – “FF08","errorMessage":"Payment Reference is invalid"
  - "amount"
    - Provide “,” e.g. 12,09 – “PA02","errorMessage":"Amount value is missing or not a valid number”
    - Provide less than 1 e.g. 0.5 - "AM06","errorMessage":"Specified transaction amount is less than agreed minimum"
    - Provide 3 decimals e.g. 100.777 – “PA02","errorMessage":"Amount value is missing or not a valid number”
  - "payeeAlias"
    - ~~Provide a number that does not match the value in the certificate – “PA01","errorMessage":"Parameter is not correct."~~
      Not for MSS!
  - "payerAlias"
    - Provide a too long or short number - "BE18","errorMessage":"Payer alias is invalid"
  - "currency"
    - Provide another value than “SEK” - :"AM03","errorMessage":"Invalid or missing Currency"
- Retrieve payment/refund result (callback confirmation)
  - Provide an invalid ID - HTTP/1.1 404 Not Found
- Refund
  - "payerPaymentReference"
    - Provide too long reference – “FF08","errorMessage":"Payment Reference is invalid"
    - Provide NULL – “FF08","errorMessage":"Payment Reference is invalid"
  - "originalPaymentReference"
    - This value is taken from the Payment Request callback element “paymentReference”
    - Use a value that is not valid i.e. change a value - "RF02","errorMessage":"Original Payment not found or original payment is more than than 13 months old"
  - "payerAlias"
    - ~~Provide a number that does not match the value in the certificate – “PA01","errorMessage":"Parameter is not correct."~~
      Not for MSS!
  - "amount"
    - Provide an amount that is greater than the original payment - "RF08","errorMessage":"Amount value is too large or amount exceeds the amount of the original payment minus any previous refunds"
      Not for MSS!
    - Provide “,” e.g. 12,09 – “PA02","errorMessage":"Amount value is missing or not a valid number”
    - Provide less than 1 e.g. 0.5 - "AM06","errorMessage":"Specified transaction amount is less than agreed minimum"
    - Provide 3 decimals e.g. 100.777 – “PA02","errorMessage":"Amount value is missing or not a valid number”

- "currency"
    - Provide another value than “SEK” - :"AM03","errorMessage":"Invalid or missing Currency"