Version 2.0

Merchant Swish Simulator

Guide

2019-03-08

swish®

Table of Contents

1. Introduction

1.1 Background

The Merchant Swish Simulator (MSS) is a test tool to offer a way for merchants to test the different available API operations and to verify the format and content of the API calls to Swish without further integration with other system components. This is done by returning a mocked response. Furthermore, the MSS can be used to test different error scenarios by specifying which error code the API should return in the request.

1.2 Prerequisites

1.2.1 TLS certificates

In order to communicate with the Swish server, you need to use the Swish client TLS certificate in the file “Swish Merchant Test Certificate 1231181189.p12”. The file contains the Swish Merchant Test Certificate together with the complete chain of trust and the private key. The password for the private key is “swish”. The test certificate is used together with the Swish number in the file name, e.g. 123 118 11 89.

It is necessary to provide the Swish client TLS certificate together with all CA certificates up to the Swish Root CA in order to correctly set up a TLS session with the Swish API.

It is recommended to require verification of the Swish server TLS certificate and not to ignore this verification, in case your server allows you to disable server certificate verification. The Swish server TLS certificate is issued under the same Swish Root CA as the Swish Merchant Test Certificate (i.e. “Test Swish Root CA v1 Test”). The Swish Root CA certificate is available in the file mentioned above but also in a separate file “Test Swish Root CA v1 Test.pem”.

1.2.2 TLS for the callback endpoint

The callback endpoint has to use HTTPS on port 443 and it is highly recommended to use IP filtering as well. For the callback, Swish will be acting as client and the merchant server is acting as server. Swish will validate the merchant callback server TLS certificate against a list of commonly recognized CAs.

2. Examples

2.1 Create Payment Request

Merchants can create payment requests for both E-Commerce and M-Commerce using MSS. Once MSS receives a "Create Payment Request" call from the merchant, there are two answers that will be returned from MSS. The first answer is synchronous, the second one is asynchronous. Please note the following:

1. MSS will directly return a "Payment request created" response to the merchant. In the case of M-Commerce, the response will also contain a Token, which is unique for each payment request.
2. MSS will perform a callback to the merchant's callback URL after some delay, the default is five seconds. This delay is only configurable by Getswish.

In order to create a payment request, a Payment Request object needs to be posted to the following URL:

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

For more information on performing a "Create Payment Request" operation, see the Swish Merchant Integration Guide available at: https://developer.getswish.se/merchants/.

Example create request (e-commerce)

```
curl -v --data '{ "payeePaymentReference": "0123456789", "callbackUrl":
"https://example.com/api/swishcb/paymentrequests", "payerAlias":
"4671234768", "payeeAlias": "1231181189", "amount": "100", "currency":
"SEK", "message": "Kingston USB Flash Drive 8 GB" }' -H "Content-Type:
application/json" POST https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/paymentrequests --cert "Swish Merchant Test Certificate
1231181189.p12:swish" --cert-type p12 --cacert "Swish TLS Root CA.pem"
```

Example create response (e-commerce)

```
< HTTP/1.1 201
< Location: https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/paymentrequests/DFEC8B87CFC74882BCC832DA6B125332
< Server: nginx/1.12.1
< Connection: keep-alive
< Content-Length: 0
< Date: Fri, 04 Jan 2019 08:28:17 GMT
<
* Connection #1 to host mss.cpc.getswish.net left intact
```

2.1.1 Simulating an error

With the MSS it's possible to test out different error scenarios by specifying which error code the API should return. This can be done by setting the value for the message property in the Payment Request object to an error code. For a list of possible error codes to set, see the Swish Merchant Integration Guide.

Example create request (error BE18)

```
curl -v --data '{ "payeePaymentReference": "0123456789", "callbackUrl":
"https://example.com/api/swishcb/paymentrequests", "payerAlias":
"4671234768", "payeeAlias": "1231181189", "amount": "100", "currency":
"SEK", "message": "BE18" }' -H "Content-Type: application/json" POST
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/paymentrequests --cert
"Swish Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --
cacert "Swish TLS Root CA.pem"
```

Example create response (error BE18)

```
< HTTP/1.1 422
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Mon, 07 Jan 2019 15:35:11 GMT
```

```
<
* Connection #1 to host mss.cpc.getswish.net left intact
[{"errorCode":"BE18","errorMessage":"","additionalInformation":"Payer
alias is invalid"}]
```

2.2 Retrieve Payment Request

Merchants can retrieve a Payment Request by sending a GET request to the following URL:

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

This URL is returned in the “Location” header defined in the response of a Create Payment Request.

Once MSS receives a “Retrieve Payment Request” call from the merchant, MSS performs a callback to the merchant’s callback URL with the status of the Payment Request. It is possible to simulate all possible values of the Payment status by providing the Payment request id of a call made earlier that leads to that outcome.

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.

Example retrieve request

```
curl -v "Content-Type: application/json" GET
https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/paymentrequests/5D59DA1B1632424E874DDB219AD54597 --cert
"Swish Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --
cacert "Swish TLS Root CA.pem"
```

Example retrieve response

```
< HTTP/1.1 200
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Fri, 04 Jan 2019 09:00:29 GMT
<
* Connection #1 to host mss.cpc.getswish.net left intact
{"id":"5D59DA1B1632424E874DDB219AD54597","payeePaymentReference":"0123456
789","paymentReference":"1E2FC19E5E5E4E18916609B7F8911C12","callbackUrl":
"https://example.com/api/swishcb/paymentrequests","payerAlias":"467123476
8","payeeAlias":"1231181189","amount":100.00,"currency":"SEK","message":"
Kingston USB Flash Drive 8 GB","status":"PAID","dateCreated":"2019-01-
02T14:29:51.092Z","datePaid":"2019-01-
02T14:29:55.093Z","errorCode":null,"errorMessage":""}
```

2.2.1 Simulating an error

Errors can also be simulated for the “Retrieve Payment Request” operation. For example, if a Payment Request is created with the error code BANKIDCL as described previously, the Retrieve Payment Request

will produce the following output:

Example retrieve request

```
curl -v "Content-Type: application/json" GET
https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/paymentrequests/0961895AF5F14AD88850A2D725402CFC --cert
"Swish Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --
cacert "Swish TLS Root CA.pem"
```

Example simulated error retrieve response

```
< HTTP/1.1 200
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Tue, 08 Jan 2019 08:12:16 GMT
```

```
<
* Connection #1 to host mss.cpc.getswish.net left intact
{"id":"0961895AF5F14AD88850A2D725402CFC","payeePaymentReference":"0123456
789","paymentReference":null,"callbackUrl":"https://example.com/api/swish
cb/paymentrequests","payerAlias":"4671234768","payeeAlias":"1231181189","
amount":100.00,"currency":"SEK","message":"BANKIDCL","status":"ERROR","da
teCreated":"2019-01-
08T08:11:55.737Z","datePaid":null,"errorCode":"BANKIDCL","errorMessage":"
"}
```

2.3 Create Refund

Merchants can send a “Create Refund” call to the MSS. Once MSS receives a “Create Refund” call from a merchant, it validates that the request refers to an available Payment Request with status PAID. If validation passes, there are two answers that will be returned from MSS. The first answer is synchronous, the second one is asynchronous. Please note the following:

1. MSS will directly return a “Refund created” response to the merchant.
2. MSS will perform a callback to the merchant’s callback URL after some delay, the default is five seconds. This delay is only configurable by Getswish.

In order to create Refund, a Refund Request object needs to be posted to the following URL:

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

For more information on performing a “Create Refund” operation, see the Swish Merchant Integration Guide available at: https://developer.getswish.se/merchants/.

When creating a refund with MSS, you may specify a value for the payeeAlias object.

Example create request

```
curl -v --data '{ "originalPaymentReference":
"5D59DA1B1632424E874DDB219AD54597", "callbackUrl":
"https://example.com/api/swishcb/paymentrequests", "payerAlias":
"1231181189", "amount": "100", "currency": "SEK", "message": "Refund for
Kingston USB Flash Drive 8 GB" }' -H "Content-Type: application/json"
POST https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds --cert
"Swish Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --
cacert "Swish TLS Root CA.pem"
```

Example create response

```
< HTTP/1.1 201
< Location: https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/refunds/C82BD1C038D4419F9126753AC2734989
< Server: nginx/1.12.1
< Connection: keep-alive
< Content-Length: 0
< Date: Tue, 08 Jan 2019 09:26:50 GMT
<
* Connection #1 to host mss.cpc.getswish.net left intact
```

2.3.1 Simulating an error

With the MSS it’s possible to test out different error scenarios by specifying which error code the API should return. This can be done by setting the value for the message property in the Refund object to an error code. For a list of possible error codes to set, see the Swish Merchant Integration Guide.

Example create request (error ACMT07)

```
curl -v --data '{ "originalPaymentReference":
"5D59DA1B1632424E874DDB219AD54597", "callbackUrl":
```

```
"https://example.com/api/swishcb/paymentrequests", "payerAlias":
"1231181189", "amount": "100", "currency": "SEK", "message": "ACMT07" }'
-H "Content-Type: application/json" POST
https://mss.cpc.getswish.net/swish-cpcapi/api/v1/refunds --cert "Swish
Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --cacert
"Swish TLS Root CA.pem"
```

Example create response (error ACMT07)

```
< HTTP/1.1 422
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Tue, 08 Jan 2019 09:28:13 GMT
<
* Connection #1 to host mss.cpc.getswish.net left intact
[{"errorCode":"ACMT07","errorMessage":"Payee alias not
enrolled","additionalInformation":null}]
```

2.4 Retrieve Refund

Merchants can perform a Retrieve Refund operation by sending a GET request to the following URL:

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

This URL is returned in the “Location” header defined in the response of a Create Refund operation.

Example retrieve request

```
curl -v "Content-Type: application/json" GET
https://mss.cpc.getswish.net/swish-
cpcapi/api/v1/refunds/A16863C914EE4A6E8FF373D5603C6913 --cert "Swish
Merchant Test Certificate 1231181189.p12:swish" --cert-type p12 --cacert
"Swish TLS Root CA.pem"
```

Example retrieve response

```
< HTTP/1.1 200
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Tue, 08 Jan 2019 10:21:33 GMT
<
* Connection #1 to host mss.cpc.getswish.net left intact
{"id":"A16863C914EE4A6E8FF373D5603C6913","paymentReference":"181BE5E44AA1
41D19FC5363D8DF5F2DE","payerPaymentReference":"","originalPaymentReferenc
e":"5D59DA1B1632424E874DDB219AD54597","callbackUrl":"https://example.com/
api/swishcb/paymentrequests","payerAlias":"1231181189","payeeAlias":"test
123","amount":100.00,"currency":"SEK","message":"Refund for Kingston USB
Flash Drive 8 GB","status":"PAID","dateCreated":"2019-01-
08T08:42:13.523Z","datePaid":"2019-01-
08T08:42:21.709Z","errorMessage":null,"additionalInformation":null,"error
Code":null}
```

3. Testing Tips

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."
  - "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"
- 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 13 months old"
  - "payerAlias"
    - Provide a number that does not match the value in the certificate – “PA01","errorMessage":"Parameter is not correct."
  - "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"
      - Note – in production the payment balance is updated after each refund but in this simulator only the original amount is used in the validation
    - 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"