Version 1.0

Swish QR Codes for Terminals

Integration Guide

2018-03-29

swish®

Table of Contents

Introduction

Document purpose

This document describes the integration of the Swish QR codes for payments in stores, with the merchant's systems. It is aimed at anyone who is tasked with connecting to the Swish API to perform QR code payments.

For more in-depth information about related topics, such as the e-commerce and m-commerce flows, and for information on how to enroll to Swish for merchants, please see https://developer.getswish.se/merchants/ and technical details https://developer.getswish.se/qrterminal/1-introduction/

Swish for merchants overview

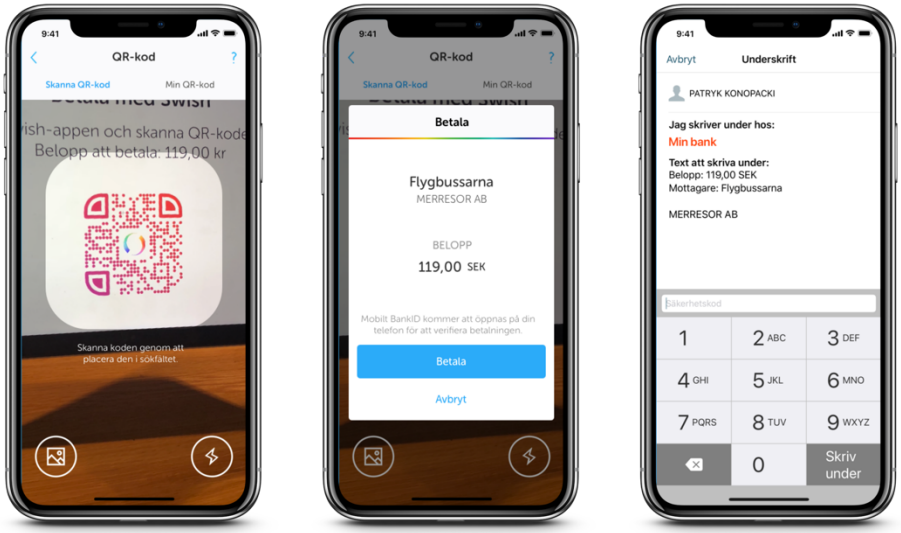
Millions of consumers in Sweden are familiar with Swish. It's the standard way to handle person-to-person transactions, between friends, colleagues and family. It is recognized as an easy-to-use, reliable and secure way to transfer money.

The Swish for merchants (Swish för handel) service extends this familiar way of doing transactions to payments on the web, in apps and in stores, and also supports refunds.

Use Case

The Swish QR codes for terminals flow is used for payments in stores.

A customer who chooses to use Swish for payment, uses the Swish app to scan a QR code for the purchase, that is presented by the cashier. This initiates the payment. When the payment is done, the cashier is notified and informs the customer of the result. The customer can view the payment in the events screen in the app.

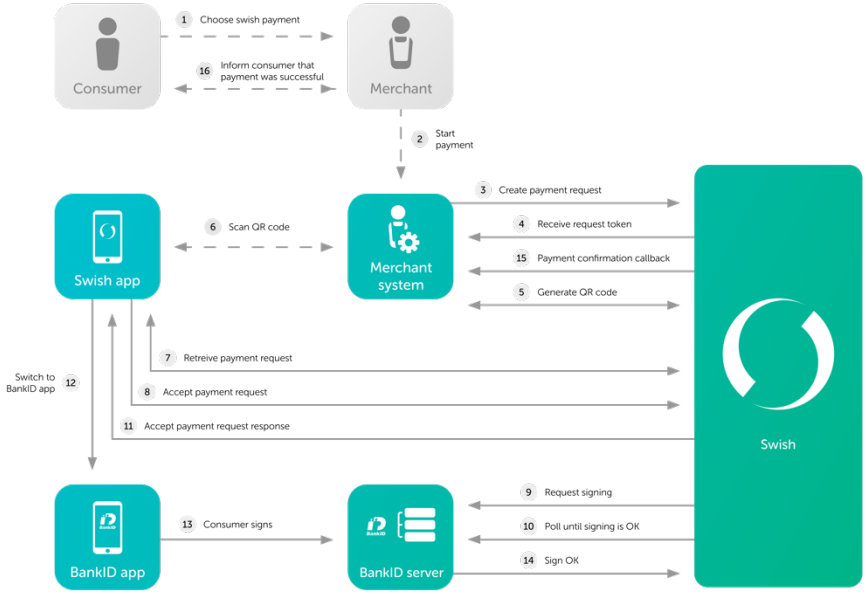


API Description

Swish QR codes for terminals

There are three main flows for commerce payment requests: m-commerce, e-commerce and QR codes (q-commerce). M-commerce and e-commerce are described in Guide Swish API. This document describes the q-commerce case.

The main difference between e-commerce and m-commerce on one hand, and q-commerce on the other hand, is that for q-commerce, there is no merchant app that communicates directly with the Swish app or a webstore. Q-commerce is used for payment in stores, where the cashier presents a QR code to the consumer, who opens the Swish app and scans the QR code to initiate the payment.



This is a step-by-step description of the q-commerce payment flow, with the merchant integration points in **bold**:

1. The consumer chooses to pay with Swish in the store.
2. The cashier starts the payment in the merchant system.
3. **The merchant system sends a payment request to the Swish system** using the API. The transaction contains data such as: amount, currency, merchant (payee) payment reference and an optional message to the consumer.
4. **The merchant system receives a Request Token**.
5. **The merchant system generates a QR-code** for the received Request Token, either using the Swish QR-generator or creating it locally.
6. The QR code is displayed to the consumer on the point-of-sale terminal.
7. The consumer starts the Swish app and scans the QR-code. The Swish app displays the payment request to the consumer.
8. The consumer clicks Pay (”Betala”) and the Mobile BankID app opens automatically for signature of the payment transaction.
9. The consumer confirms the payment transaction by signing with the Mobil BankID using his/her password.
10. The amount is transferred in real-time from the consumer’s account to the merchant’s account.
11. The consumer stays in the Swish app.
12. **The merchant receives a confirmation of the successful payment**, via a callback from the Swish system to the merchant system.
13. The cashier informs the consumer about the outcome of the payment.

14. The consumer can view the payment in the events section (“Händelser”) as any other payment in the Swish app.

For brevity, “behind the scenes” interactions, for example the Swish system’s interaction with BankID and the consumer’s bank, are not included in the steps listed above.

The integration points that need to be implemented in the merchant system are:

- Create a payment request. This initiates the payment.
- Retrieve payment request token. The token identifies the payment request and is encoded in the QR code.
- Generate a QR code. The payment request token is embedded in the QR code, which is scanned by the customer using the Swish app.
- Receive a confirmation callback when the payment is finished.

Create payment request

A payment request is created by posting the relevant information to the Swish for merchants API:

```
POST /api/v1/paymentrequests
```

Example:

```
curl -v --request POST https://swicpc.bankgirot.se/swish-cpcapi/api/v1/paymentrequests \
    --header "Content-Type: application/json" --data @- << !
{
    "payeePaymentReference": "0123456789",
    "callbackUrl": "https://example.com/api/swishcb/paymentrequests",
    "payeeAlias": "1234760039",
    "amount": "100",
    "currency": "SEK",
    "message": "Kingston USB Flash Drive 8 GB"
} !

< HTTP/1.1 201 Created
< Location: https://swicpc.bankgirot.se/swish-
cpcapi/api/v1/paymentrequests/AB23D7406ECE4542A80152D909EF9F6B
< PaymentRequestToken: umP7Eg2HT_OUId8Mc0FHPCxhX3Hkh4qI
```

Retrieve payment request

```
GET api/v1/payment-requests/{id}
```

Example:

```
curl -v --request GET https://swicpc.bankgirot.se/swish-
cpcapi/api/v1/paymentrequests/AB23D7406ECE4542A80152D909EF9F6B

< HTTP/1.1 200 OK
{
    "id": "AB23D7406ECE4542A80152D909EF9F6B",
```

```
    "payeePaymentReference": "0123456789",
    "paymentReference": "6D6CD7406ECE4542A80152D909EF9F6B",
    "callbackUrl": "https://example.com/api/swishcb/paymentrequests",
    "payeeAlias": "1231234567890",
    "amount": "100",
    "currency", "SEK",
    "message": "Kingston USB Flash Drive 8 GB",
    "status": "PAID",
    "dateCreated": "2015-02-19T22:01:53+01:00",
    "datePaid": "2015-02-19T22:02:12+01:00"
}
```

Generate QR code

There are two ways to generate a QR code:

- using the Swish QR-generator, or
- generating it locally according to guidelines provided by Swish.

What is described in this section is how to use the Swish QR-generator, but if you expect to generate large volumes of QR codes, do not hesitate to contact us.

The QR code to display to the customer is generated by posting the payment request token to the Swish QR-generator API.

The string represented by the QR code will be the token, prefixed with the capital letter D. So in the current example, the token is `umP7Eg2HT_OUId8Mc0FHPCxhX3Hkh4qI`, and the QR code will contain the string `DumP7Eg2HT_OUId8Mc0FHPCxhX3Hkh4qI`.

The QR-generator API is described in detail in the QR Code Integration API document, see https://developer.getswish.se/qr/

```
POST /api/v1/commerce
```

Example:

```
curl -v --request POST https://www.swish.bankgirot.se/qrg-swish/api/v1/commerce --header
"Content-Type: application/json" --data @- << !
{
    "format": "png",
    "size": 300,
    "token": "umP7Eg2HT_OUId8Mc0FHPCxhX3Hkh4qI"
}
!
```

The returned QR code, containing the string `DumP7Eg2HT_OUId8Mc0FHPCxhX3Hkh4qI`:

Callback

Swish will make a callback HTTPS POST request with the Payment Request Object JSON as payload, to the Callback URL supplied in the Create Payment Request operation when either of the following events (as set in the `status` property of the Payment Request Object) happens:

- PAID - The payment was successful.
- DECLINED - The payer declined to make the payment.
- ERROR - Some error occurred, like payment was blocked, payment request timed out etc. See the list of error codes for all potential error conditions.

A payment request has to be accepted or declined by the consumer within three (3) minutes. When the time has elapsed an ERROR status is returned to the Callback URL. If the consumer accepts the payment request a status is returned to the Callback URL within 12 seconds.

The callback endpoint has to use HTTPS, and IP filtering is highly recommended as well. It is up to the merchant to make sure the endpoint is available.

Swish will only make the callback request once. If the merchant has not received a callback response after the timeout, the merchant can choose to retrieve the payment request to check the result. Swish will always try to make a callback request before the timeout period, but if it times out, a timeout callback is sent with status ERROR and the error code value TM01.

API Reference

Date format

Dates are represented using the ISO 8601 date format. Since the Swish servers create these fields, and the servers are located in Sweden, the timezone used is CET, which is UTC+01:00 or UTC+02:00, depending on whether it is Central European Summer Time (CEST) or not. See the code examples for samples.

Payment request

The payment request object is used in all three payment request operations: Create, Retrieve and Callback.

Legend:

- M – Mandatory input for Create operation
- O – Optional input for Create operation
- R – Response parameter, should not be specified for Create operation

Property	Type		Description
`id`	*string*	R	Payment request ID
`payeePaymentReference`	*string*	O	Payment reference of the payee, which is the merchant that receives the payment. This reference could be order id or similar.
`paymentReference`	*string*	R	Payment reference from the bank, of the payment that occurred based on the Payment request. Only available if status is PAID.
`callbackUrl`	*string*	M	URL that Swish will use to notify caller about the outcome of the Payment request. The URL has to use HTTPS.
`payerAlias`	*string*	O	Not relevant for q-commerce.

payeeAlias	*string*	M	The Swish number of the payee, that is the merchant.
amount	*string*	M	The amount of money to pay. The amount cannot be less than 1 SEK and not more than 999999999999.99 SEK. A valid value has to be all digits, or all digits, a period and two decimal digits.
currency	*string*	M	The currency to use. SEK is the only supported value.
message	*string*	O	Merchant supplied message about the payment/order. The allowed characters are the letters a-ö, A-Ö, the digits 0-9 and the special characters :;.,?!()”.
status	*string*	R	The status of the transaction. Possible values: CREATED, PAID, DECLINED, ERROR.
dateCreated	*string*	R	The time and date when the payment request was created.
datePaid	*string*	R	The time and date when the payment request was paid. Only set if the status is PAID.
errorCode	*string*	R	A code indicating what type of error occurred. Only applicable if the status is ERROR.
errorMessage	*string*	R	A descriptive error message in English, indicating what of the error occurred. Only set if the status is ERROR.
additionalInformation	*string*	R	Additional information about the error. Only set if the status is ERROR.

Create payment request

```
POST /api/v1/paymentrequests
```

HTTP status codes:

Error code	Description
`201 Created`	Returned when a payment request was successfully created. Will return a Location header and 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 merchant's Swish number.
`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.

Potential error codes, returned in Error objects when validation fails, that is HTTP status code 422 is returned:

Code	Description
`FF08`	`paymentReference` is invalid
`RP03`	Callback URL is missing or does not use HTTPS
`RP01`	Missing Merchant Swish Number
`PA02`	`amount` value is missing or not a valid number
`AM06`	Specified `amount` is less than agreed minimum
`AM02`	Specified `amount` value is too large

AM03	Invalid or missing currency
RP02	Incorrectly formatted message
ACMT01	Counterpart is not activated

Retrieve payment request

```
GET api/v1/payment-requests/{id}
```

Potential HTTP status codes returned:

HTTP status codes:

Status code	Description
200 OK	Returned when Payment request was found. Will return Payment Request Object.
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.
404 Not found	Returned when the Payment request was not found or it was not created by the merchant. 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.

Error objects

An array of error objects is returned when a Create payment request operation fails with status code 422.

Property	Type	Description
errorCode	*string*	A code indicating what sort of error occurred.
errorMessage	*string*	A human-readable error message in English, describing the error that occurred.

`additionalInformation`	*string*	Any additional information about the error.

Example of an array of error objects:

```
 [{
    "errorCode": "PA02",
    "errorMessage": "Amount value is missing or not a valid number",
    "additionalInformation": ""
 },{
    "errorCode": "AM03",
    "errorMessage": "Invalid or missing Currency",
    "additionalInformation": ""
 },{
    "errorCode": "RF08",
    "errorMessage": "Amount value is too large or amount exceeds the amount
of the original payment minus any previous refunds",
    "additionalInformation": "100.00"
}]
```