

Swish QR Codes for terminals

Integration Guide

Table of contents

1. Introduction	3
1.1 Document purpose	3
1.2 Swish Commerce overview	3
1.3 Use case	4
2. Swish QR codes for terminals	5
2.1 Generating a QR code	6
2.1.1 API Description.....	7
2.1.2 Example.....	7

1. Introduction

1.1 Document purpose

This document describes the integration of 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 the Swish Commerce service, please see the Swish Merchant Integration Guide available at <https://developer.getswish.se/merchants/>.

1.2 Swish Commerce overview

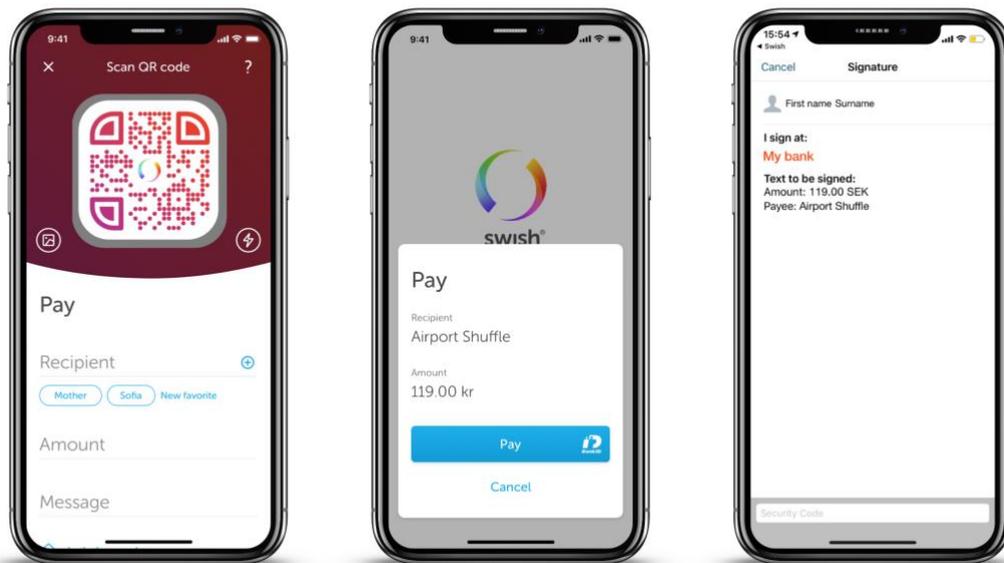
Millions of consumers in Sweden are familiar with Swish. It's the standard way to handle person-to-person transactions, between friends, colleagues and family. It is recognized as an easy-to-use, reliable and secure way to transfer money.

The Swish Commerce service extends this familiar way of doing transactions to payments on the web, in apps and in stores, and also supports refunds.

1.3 Use case

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

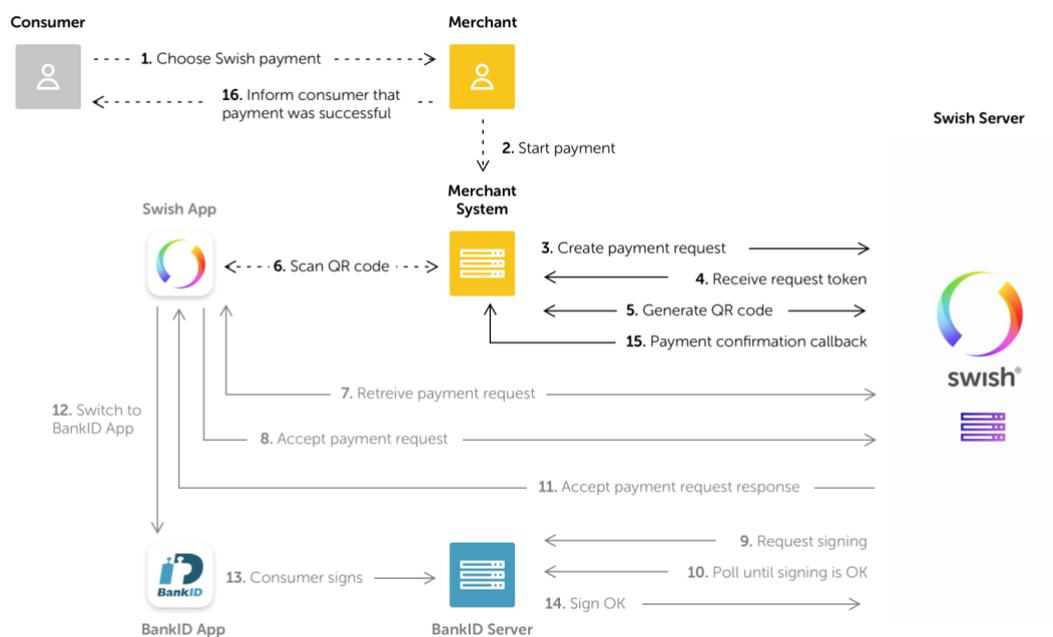
When a customer who chooses to use Swish for payment, the cashier initiates a Swish payment which displays the QR code in the terminal. The customer uses the Swish app to scan a QR code for the purchase, which initiates the payment. The customer can choose to accept or cancel the payment in the Swish app. 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 view in the app.



2. 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 the Merchant Integration Guide. This document describes the q-commerce case.

Technically, the q-commerce flow is a type of m-commerce flow where the payment request token is delivered to the Swish app via a QR code instead of an app call. The q-commerce flow is typically used in point-of-sale terminals where there is no merchant app or website that communicates directly with the Swish app. When Q-commerce is used for payment in stores, the cashier presents a QR code to the consumer, who needs to open their Swish app and scan 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 creates an m-commerce payment request in the Swish system** using the Swish Commerce 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 Payment Request Token.**
5. **The merchant system generates a QR-code** for the received Payment 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 and the Mobile BankID app opens automatically for signature of the payment transaction.
9. The consumer confirms the payment transaction by signing with the Mobile 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 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:

- Creating an m-commerce payment request. This initiates the payment.
- Retrieving the payment request token. The token identifies the payment request and is encoded in the QR code.
- Generating a QR code. The payment request token is embedded in the QR code, which is scanned by the customer using the Swish app.
- Receiving a confirmation callback when the payment is finished.

2.1 Generating a QR code

There are two ways of generating a QR code:

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

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

The QR code is created in two steps. The first step is to create an M-Commerce Payment Request using the Swish Commerce API. For details on how to create an M-Commerce Payment Request, see the Swish Merchant Integration Guide.

The second step is to post the PaymentRequestToken received in the Payment Request response to the Swish QR-generator API. This will return the raw binary data for the QR image.

The string represented by the QR code will be the token, prefixed with the capital letter D. So if for example the PaymentRequestToken is `umP7Eg2HT_OUIId8Mc0FHPCxhX3Hkh4qI`, the QR code will contain the string `DumP7Eg2HT_OUIId8Mc0FHPCxhX3Hkh4qI`.

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



2.1.1 API Description

Request

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

Properties

Property	Type	Required	Description
token	string	Yes	The Payment Request token to create a QR code for.
format	string	Yes	Possible values: "jpg", "png", "svg".
size	number	No	Size of the QR code. The code is a square, so width and height are the same. Not required if the format is svg.
border	number	No	Width of the border
transparent	boolean	No	Transparent background color. Does not work with jpg format.

2.1.2 Example

Example request

```
curl -v --data '{ "format": "png", "size": 300, "token":  
"umP7Eg2HT_OUIId8Mc0FHPCxhX3Hkh4qI"}' -H "Content-Type: application/json"  
POST https://mpc.getswish.net/qrg-swish/api/v1/commerce
```

Example response

The returned QR code, containing the string DumP7Eg2HT_OUIId8Mc0FHPCxhX3Hkh4qI:

