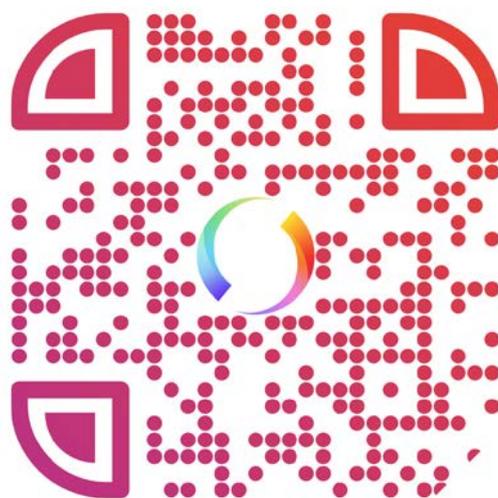


# Guide Swish QR Code specification

Integration Guide



Datum: 2017/07/07



# Table of content

1	Introduction.....	3
2	Payment flows .....	3
2.1	Swish C2B flow with QR code .....	3
3	Create QR code via getswish.se .....	4
4	Create QR codes using Swish QR code generator APIs.....	4
4.1	Available endpoints .....	4
4.1.1	Prefilled .....	4
4.1.2	Prefilled request .....	5
5	Create your own Swish QR code .....	6
5.1	Design specification – color .....	6
5.2	Design specification – black and white .....	7
6	QR code encoding specification.....	7
6.1	Swish C2B.....	8
7	Adresses.....	8
7.1	Production environment .....	8

# 1 Introduction

In the bank common Swish-app for our private consumers, it is possible to scan Swish QR-codes to fill the payment information on the “Pay-page” in the app.

There are three ways to create QR-codes to be scanned in the Swish-app:

1. Create codes using Swish QR-code generator on [www.getswish.se/qr](http://www.getswish.se/qr)
2. Create codes using APIs towards the Swish QR-code generator
3. Create the codes yourself following the description in this document in order to get the right design of and content in the code.

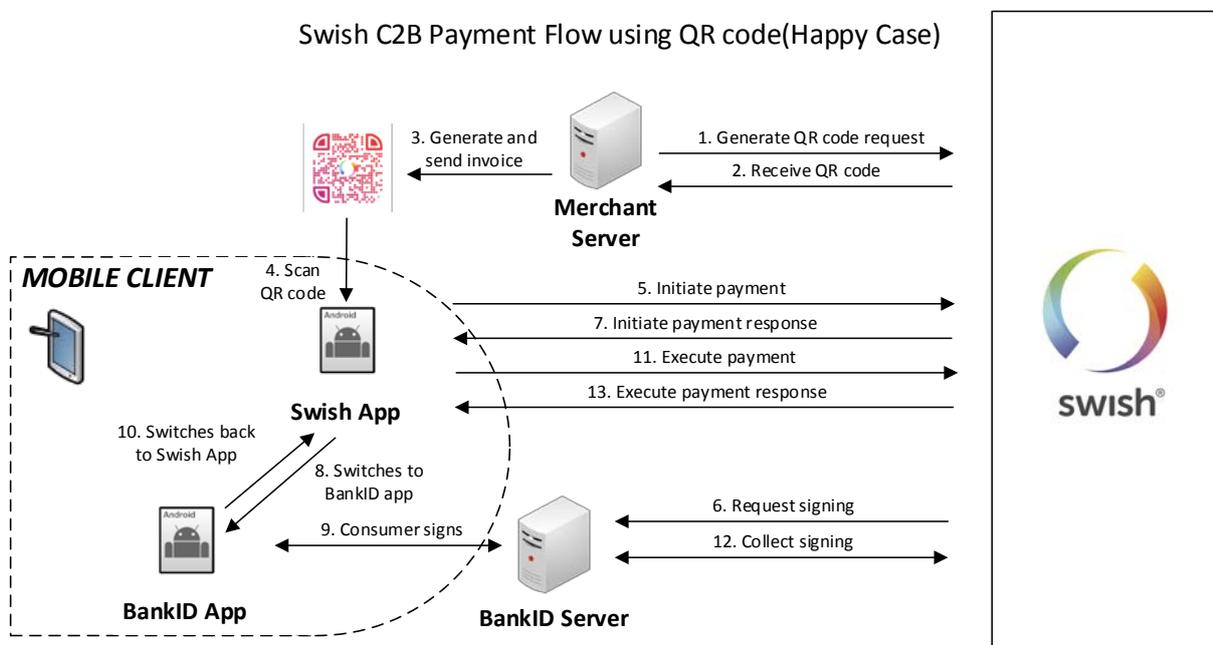
This specification describes how to create Swish QR-codes using APIs towards the Swish QR-code generator and how to create Swish QR-codes with the right design and content in the code.

## 2 Payment flows

### 2.1 Swish C2B flow with QR code

The C2B flow with Swish QR code can be used by companies with the wish to facilitate the payment flow for their consumers and to make sure that the correct information, such as reference number, is included in the payment information. A company contacts Swish to get a QR code which they show their consumer to initiate a Swish-payment. The consumer scans the code using the QR-code function in the Swish app and the information is prefilled on the “Pay-page” in the app. He/she approves the payment by pressing “Pay” and then signs the payment using Mobile BankID.

The QR code specification, see part 6 of the document, is relevant for companies willing to replace steps 1 and 2 by generating the Swish QR code themselves.



1. The company backend calls the Swish QR code generator API (see the dedicated QR code generator API, separate document).
2. The company backend receives a QR code.
3. The QR code is displayed to the consumer.



4. The consumer opens the Swish app and scans the QR code. The prefilled payment is displayed in the Swish app.
5. The consumer presses the “Pay” button.
6. A signing session is initiated with Mobile BankId.
7. Swish answers the Swish app.
8. The Swish app switches to the BankId app.
9. The consumer signs in BankId app.
10. The BankId app switches back to the Swish app.
11. The Swish app contacts Swish to execute the payment.
12. Swish checks the outcome of the signing with BankID. The money is then transferred, in real time, from the consumer’s account to the company’s account.
13. The consumer see that the payment was performed successfully in the Swish app.

The consumer can view the payment in the events section “Händelser” as a sent payment in the Swish app.

## 3 Create QR code via [getswish.se](https://getswish.se)

Getswish offers the possibility to create Swish QR-codes at [www.getswish.se/qr](https://www.getswish.se/qr).

The code can include

1. The receiving mobile or Swish-number (required)
2. The amount of the payment (optional)
3. The payment information such as reference number or order ID. (optional)

It is possible to define if number 2 and/or 3 will be editable by the payer or not.

There are marketing material which can be used to display the QR-code. For further information, please visit [www.getswish.se/qr](https://www.getswish.se/qr).

## 4 Create QR codes using Swish QR code generator APIs

Getswish offers a solution to send a request via an API to create Swish QR-codes. The code can be delivered in three different formats; JPG, PNG and SVR.

The code can include

1. The receiving Swish-number (required)
2. The amount of the payment (optional)
3. The payment information such as reference number or order ID. (optional)

It is possible to define if number 2 and/or 3 will be editable by the payer or not.

### 4.1 Available endpoints

#### 4.1.1 Prefilled

POST `/api/v1/prefilled`

The Http request body have to contain a Prefilled request.



Potential Http status codes returned:

200 OK: Returned when the QR code was successfully created. Will return a QR code response.

400 Bad Request: Returned when the Prefilled request operation was malformed.

415 Unsupported Media Type: Returned when Content-Type header is not "application/json".

500 Internal Server Error: Returned if there was some unknown/unforeseen error that occurred on the server, this should normally not happen.

#### 4.1.2 Prefilled request

The JSON payload follows the schema (shall be properly JSON serialized to avoid problems with uncommon characters in message).

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Swish pre-filled qr code generator",
  "description": "REST interface to get a QR code that the Swish app will
interpret as a pre filled code",
  "definitions": {
    "editable": {
      "description": "Controls if user can
modify this value in Swish app or not",
      "type": "object",
      "properties": {
        "editable": {
          "type": "boolean",
          "default": false
        }
      }
    },
    "swishString": {
      "type": "object",
      "properties": {
        "value": {
          "type": "string",
          "maxLength": 70
        }
      },
      "required": ["value"]
    },
    "swishNumber": {
      "type": "object",
      "properties": {
        "value": {
          "type": "number"
        }
      },
      "required": ["value"]
    }
  },
  "type": "object",
  "properties": {
    "format": {
      "enum": ["jpg", "png", "svg"]
    },
    "payee": {
      "description": "Payment receiver",
      "allOf": [
        {"$ref": "#/definitions/editable"},

```



```

        {"$ref": "#/definitions/swishString"}
    ]
},
"amount": {
    "description": "Payment amount",
    "allof": [
        {"$ref": "#/definitions/editable"},
        {"$ref": "#/definitions/swishNumber"}
    ]
},
"message": {
    "description": "Message for payment",
    "allof": [
        {"$ref": "#/definitions/editable"},
        {"$ref": "#/definitions/swishString"}
    ]
},
"size": {
    "description": "Size of the QR code. The code is a
square, so width and height are the same. Not required is the
format is svg",
    "value": "number",
    "minimum": 300
},
"border": {
    "description": "Width of the border.",
    "type": "number"
},
"transparent": {
    "description": "Select background color to be
transparent. Do not work with jpg format.",
    "type": "boolean"
}
},
"required": ["format"],
"anyOf": [
    {"required": ["payee"]},
    {"required": ["amount"]},
    {"required": ["message"]}
],
"additionalProperties": false,
"maxProperties": 5
}

```

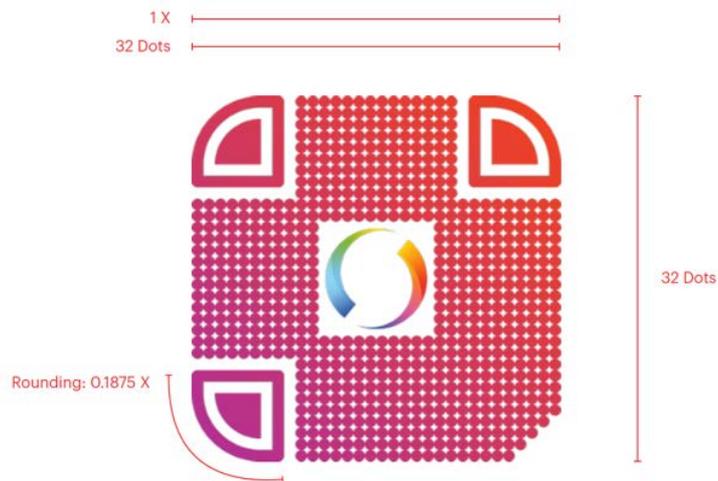
## 5 Create your own Swish QR code

### 5.1 Design specification – color



This is the form and design of the Swish QR-code. It has 3 rounded eyes in the corners.

It is designed to look coherent with the rest of Swish. It also have the Swish symbol in the middle to emphasise the Swish brand no matter where it is applied.



The color of the Swish QR code is a 45° purple to red gradient.

## 5.2 Design specification – black and white

It is possible to create Swish QR-code in black and white, following the form and design specified above and the logo must follow the [Swish trademark guidelines](#)

# 6 QR code encoding specification

This section describes how to create Swish QR-code with the set design and with the correct content in order to be possible to be scanned and used in the Swish-app.

The code can include

1. The receiving Swish-number (required)
2. The amount of the payment (optional)
3. The payment information such as reference number or order ID. (optional)

It is possible to define if number 2 and/or 3 will be editable by the payer or not.



## 6.1 Swish C2B

*C<payee>;<amount>;<message>;<lock\_mask>*

Where:

- C (capital letter) is the required prefix
- Payee is the corporate Swish Number.
- Amount is in SEK, with 2 decimals. Decimal separator is a comma.
- Message is URL encoded. It can be a reference number, Order ID or on a invoice, the OCR number.
- Lock mask controls what parameters the consumer is allowed to edit in the payment view before making the payment:
  - Decimal value of bitmask where payee has bit position 0, amount 1 and message 2.
  - True means editable, false means parameter is fixed.
  - If amount and message shall be editable we will get the mask 110 in binary form, and then mask parameter has the value  $4 + 2 + 0 = 6$
  - Will in most cases be 0
- Parameters are omitted by using empty string.
- Omitted lock mask is interpreted as value 0

I.e. if a company (1237856901) send an invoice for 100SEK with OCR number 12229445, then the content of the QR code is one of:

*C1237856901;100,00;12229445;0*

*C1237856901;100,00;12229445*

# 7 Adresses

## 7.1 Production environment

The Swish server IP address for IP filtering:  
213.132.115.94:443

Swish API URL: <https://mpc.getswish.net/qrg-swish/api/v1/prefilled>

