

Developer Documentation for Swedbank RestFX Market Order

Contents

1	Version Control.....	3
2	Introduction	3
2.1	Glossary	4
2.2	Standards.....	5
3	Sandbox Overview.....	6
4	Connecting to API.....	7
4.1	Partner registration for access to developer’s portal	7
4.2	Partner registration for production usage	9
4.3	Partner Transport Security for RestFX Market Order	10
5	Functional specification - RestFX Market orders	11
5.1	Mandatory parameters	11
5.2	Endpoints.....	12
5.3	POST order – request and response specifications.....	14
	Request – specification	14
	Response - successfully executed trade.....	17
	Response – Error generated in RestFX validation layer (A).....	19
	Response – Error generated in Swedbank’s FX trading platform (B).....	20
5.4	Market order lifecycle and example usage	22
6	RestFX API versioning guidelines.....	25
6.1	Versioning policy	25
6.2	Documentation and information about changes in API versions	26
	Breaking changes.....	26
	Non-breaking changes.....	26

1 Version Control

We encourage feedback! Please send any comments or suggestions for how this documentation could be improved to: fxautomation@swedbank.se

Version	Date	Title
1.0	2020-05-29	Launch version.
1.1	2021-10-06	Clean version Market Orders content only

2 Introduction

The purpose of this documentation is to give developers an introduction to the Foreign Exchange (FX) related services that are part of Swedbank's Open Banking suite. These offerings are gathered under the umbrella term "RestFX" derived from the abbreviation REST as in RESTful state Application Programming Interface (API) and FX as in Foreign eXchange. This document describes the best ways of API usage, explains unclear parts of the API:s and describes the API Service User onboarding process etc. Technical details on each endpoint including URL:s, mandatory request parameters, response formats and http response codes can also be found in the Swagger files on: [Open Banking Developer Portal](#).

Whilst the documentation is aimed primarily towards developers and other implementors of the API:s it can also be read as a general introduction to the services and serve as useful primer for management, business development staff etc. etc.

- **RestFX Market Order:** a simple, user friendly way to place FX orders with Swedbank. This service is strictly limited to *non-financial counterparty* (as defined in Article 2.9 of EMIR¹) customers of Swedbank AB and it is currently not offered to customers of the Swedish Savings Banks nor to customers of the Baltic parts of the Swedbank Group (Swedbank AB Lithuania, Swedbank AB Latvia, Swedbank AB Estonia). The service allows onboarded customers to place FX orders with Swedbank to be executed at the prevailing market rate including such sales margins that may be applied by the Bank from time to time.

¹<https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:32012R0648>

2.1 Glossary

The term *API Service User (ASU)* will be used throughout this document to denote the party accessing the banks RestFX API:s and consuming the services and data provided through these.

Key Terms	Definition
API Provider	Swedbank AB in this context.
API User	An API User is any person or organisation that develops web- or mobile applications that access data from an API Provider. The API User may or may not be the same as the API Service User ² .
API Service User	In a Swedbank RestFX context this term is used to denote the consumer of the RestFX API:s. I.e. this is the party requesting FX rates from the Bank, holding accounts with the Bank and the legal counterparty to the Bank in any financial transaction resulting from the use of the RestFX services.

² If the API Service User is relying on a third party such as an IT system provider for the development of the technical solution to connect to the API:s this third party would be denoted APU User.

2.2 Standards

All of Swedbank's Open Banking API:s, including the RestFX services, adhere to the standards described in ISO20022 & the [Berlin Group standard NextGenPSD2 XS2A Framework Implementation Guidelines version 1.2](#) (BGS).

The exposure of data is done through RESTful services. API consumers should respect cache policy: *VOLATILE*. For most of the API calls described in this documentation data is provided in JavaScript Object Notation (**JSON**) format. The API request and responses must use an UTF-8 character encoding.

All dates in parameters & request body are represented in ISO 8601 date or date time format e.g.:

```
ISO 8601 Date - 2019-05-30
```

```
ISO 8601 DateTime - 2019-05-30T00:00:00+00:00
```

All dates in the HTTP headers are represented as RFC 7231 Full Dates

```
Tue, 21 May 2019 14:23:49 GMT
```

3 Sandbox Overview

All the RestFX services are available also as Sandbox versions on Swedbank's Open Banking platform. The purpose of these Sandbox versions of the services is to provide easily accessible "playground" environments where the services can be tried, tested and evaluated with minimal prior onboarding and administration.

Testing the services using the Sandboxes, also gives a more comprehensive and detailed understanding of how the API:s work and serves as a good preparation for deployment of the production versions of the API:s.

Note: whilst technically and from an API specification perspective, the Sandbox versions of the API:s should be fully aligned with the production versions of the services, the Sandboxes may be based on static and/or random data i.e. the data content of responses on the Sandbox API:s do not necessarily reflect accurate market conditions nor will using these services ever result in real transactions taking place.

To use the Sandboxes (compared to production API:s) the following changes in the base URL are needed:

Environment	Base URL and API
Sandbox	https://psd2.api.swedbank.com:443/partner/sandbox/v1/fx/market-order/..
Production	https://se.psd2.api.swedbank.com/partner/v1/fx/market-order/..

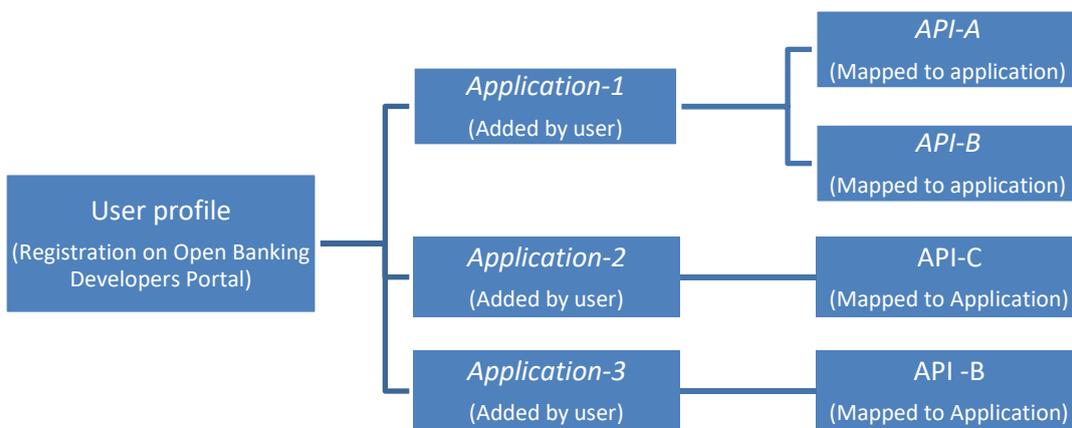
Note: Swagger files for the Sandboxes are also available on the developer's portal.

4 Connecting to API

To be able to use and connect to the API:s the, prospective, API Service Users need to follow a specified registration or “on-boarding” process and in the case of RestFX Market Order meet a few other pre-requisites before the on-boarding process can be finalised.

The first step in the process is to register in the Open Banking Developer’s portal and to create the Applications needed. Every API to be used needs to be associated with an Application. It is however possible to map several API:s to one Application and also inversely to map several Applications to one API depending on what is suitable for the specific API Service User.

Separate *Applications* are also needed for the Sandbox versions of the services.



4.1 Partner registration for access to developer’s portal

Registering for access to the developer’s portal allows the user to browse the published API:s and freely evaluate and test the Sandbox version of the service. The Sandbox version of the service can be tested using the Swagger files published in the Developers portal or using any off-the-shelf software such as Postman (www.postman.com) alternatively using a proprietary application developed for this purpose.

- Register in the [Open Banking Developer portal](#) by providing a correct email address. After few minutes an email with a link to further steps to finalize the registration and create a profile on the portal will arrive. Please safekeep the username and follow best practices for password creation and management. Unauthorized use of this account may cause damage.
- In Open Banking Developer’s portal create an application [Publish](#) → [Apps](#) → [Add application](#) for the Indicative Rates service and assign the Indicative Rates API to this Application. Different applications (app-id’s) must be used for the for the sandbox and production environment.
- Assign needed API’s to application.

If a CA certificate is provided Sandbox will do a validation with mutual Transport Layer Security (TLS).

4.2 Partner registration for production usage

To get access to the production versions of the RestFX API there are a few more steps to complete but the basic process is the same. For any advice or assistance with these preparatory steps please contact: fxautomation@swedbank.com.

Prerequisites and preparations

A. Swedbank customer

As stated⁴ the RestFX Market Order will only be made available to customers of Swedbank AB.

B. FX Trade

Swedbank customers need to be FX Trade customers⁵. FX Trade is free of charge and allows customers to place orders and trade FX with the Bank through a user-friendly web based Graphic User Interface (GUI).

C. RestFX Market Order customer contract (“Villkor för Valutahandel RestFX Market Order”)

The RestFX Market Order customer contract needs to be signed.

Note: The registration process on the Open Banking page may be initiated even before these pre-requisites have been met but cannot be finalized until they are.

To register for the production version of the service follow the steps outlined below:

- Register in the [Open Banking Developer’s portal](#) as described in section 4.1.
- In Open Banking Developer’s portal create an application [Publish](#) → [Apps](#) → [Add application](#) for the Market Order service and assign the Market Order API to this Application. Please note that the CA certificate should be attached in this step.
- Send an e-mail to: fxautomation@swedbank.com with a request to have the app-id obtained in step 2 activated for the RestFX Market Order service. Once the request is approved by the bank the service is open for use.

Register Public key for your certificate (CA issued TLS/SSL certificate) to achieve mTLS. These certificates will be validated using Online Certified Status Protocol (OCSP).

After these steps are finished the Swedbank Open bank API can be used. It is strongly advised to perform API smoke test & piloting in production before opening it for wide usage.

We encourage providing feedback on documentation and suggestions for improvements to: fxautomation@swedbank.com.

⁴ Legal persons currently not banking with Swedbank need to pass the customer due diligence screening including the Know Your Customer (KYC) process to become Swedbank customers.

⁵ FXTrade is Swedbank’s FX trading portal. More information can be found on: <https://swedbank.se/foretag/affarer-med-utlandet/produkter-och-tjanster/fx-trade.html> (Swedish only).

⁶ Registration on the Open Banking Developers Portal needs to be done only once.

4.3 Partner Transport Security for RestFX Market Order

The communication between the API Service User and the Bank is always secured by using **TLS version 1.2** or higher if the API Service User provides the necessary certificate, this is however not a strict requirement.

To ensure secure communication between Partner and Swedbank and for Partner identification Swedbank relies on the following qualified certificates:

- CA issued TLS/SSL certificate.

A qualified digital certificate is a public key certificate issued by a Certificate authority that ensures the authenticity and data integrity of an electronic signature and its accompanying message and/or attached data. This assures a link between the cryptographic keys used to secure the transaction and the entity the keys belong to. CA certificates are used to achieve Transport Layer Security (TLS) on the transport level.

5 Functional specification - RestFX Market orders

The Market Order service consists of a number of endpoints. The service requires authentication for the API Service User (see section 4.4 for more on authentication) i.e. in addition to the *app-id* parameter having been obtained in the on-boarding process when choosing to add this application in the developer's portal the user needs to pass a valid and previously approved Qualified Website Authentication Certificate (QWAC). In addition to the *app-id* parameter an *x-request-id* header needs to be passed in the request. This header is the API Service Users own unique identifier for this specific request and is not validated in the Open Banking layer. This service is as previously stated open to customers of the bank only.

5.1 Mandatory parameters

The API Service User needs to pass an *Application* ID (*app-id*) and a *Request* ID (*x-request-id*) with every request made to the RestFX services. This goes for all endpoints of this service.

Name	Category	Description
app-id	Parameter ¹⁰	<p>A unique identifier generated by the Open Banking platform. The <i>app-id</i> or <i>API Key (Client ID)</i> is created when the API Service User goes through the process of adding an <i>Application</i> in the developer's portal and assigning one of the available API:s to that <i>Application</i>. I.e. an <i>Application</i> ID uniquely identifies an API Service User and one specific "use case" of an API for that specific API Service User.</p> <p>The <i>app-id</i> is a string of 33 alphanumeric characters e.g. I479v6f9c02e9e3b5980939a819411abcc</p>
x-request-id	Header	<p>This is the API Service Users own identifier for a specific request on an endpoint and is always returned in the response (as long as the request was parsable and at least technically validated). The <i>x-request-id</i> allows the API Service User to create and follow an intact audit-trail from request to response.</p>

Examples

The *app-id* is passed as a parameter forming part of the URL as in the example below:

../indicative-rate/rate?currencyPair=EURSEK&app-id=I479v6f9c02e9e3b5980939a819411abcc

¹⁰ Part of the endpoint URL when making a request.

5.2 Endpoints

The best way to understand fully how the endpoints work is to test and evaluate them using the Sandbox Swagger interfaces. Below is a high-level description of the different endpoints:

Endpoint	Type	Description
currencypairs	GET	<p>Lists all currency-pairs supported on this service. Any currency-pair not found in this list (including inverted pairs from the list) will result in an error message.</p> <p>E.g. whilst HUFSEK is supported in the current version, HUFDKK is not and neither is the inverted pair SEKHUF.</p> <p>Note: the list returned by this endpoint contains the currency-pairs <u>technically</u> supported on this RestFX service. Whether or not a customer has the currency accounts and the rights to trade these specific currencies, and thus currency-pairs, is a different matter.</p> <p>If the customer wishes to trade a currency-pair listed on this service but is unable to do so (s)he should contact openbanking@swedbank.com¹¹ or the assigned sales representative from the bank.</p>
tenors	GET	<p>List all standard tenors supported by the service. The tenor defines the expected settlement (delivery) date of the trade being executed.</p> <p>E.g. TD equates to same-day settlement (available until 17:00 CET only), TM equates to settlement on the next following good bank day¹² for both the currencies involved and SP equates to settlement two good bank days later. The service also supports longer tenors (W = Week, M = Month, Y = Year) up to a maximum of settlement one year after the trade date.</p> <p>When placing an order to trade (via the POST orders endpoint) the customer is free to use either a tenor specified in this list (e.g. "SP") <u>or</u> specify an exact settlementDate (e.g. "2020-05-07"), what is commonly referred to as a "broken tenor".</p> <p>The difference between the two alternatives is that a specified Tenor (with the exception of TD) is always deferred to the first good bank day following the Tenor specified and is thus always accepted by the bank. When an exact date is specified the bank will validate if this specific date is a good bank day for both of the currencies involved in the trade and if that is not the case the order will be rejected.</p> <p>Note: the list returned by this endpoint contains the Tenors <u>technically</u> supported on the RestFX service. Whether or not the customer has the rights to trade these specific Tenors is a different matter.</p>

¹¹ Please refer to <https://www.swedbank.com/openbanking/report-an-issue.html> for valuable advice on what information to provide when contacting the Open Banking support. This will speed up processing significantly.

¹² Good bank day = day when retail banks are open to process transaction in that currency zone i.e. in practice non-bank holidays.

		If the customer wishes to trade a Tenor listed on this service but is unable to do so (s)he should contact openbanking@swedbank.com ¹¹ or the assigned sales representative from the bank.
orders	POST	<p>Place an FX order. Takes an order instruction and returns either the details of the trade having been executed for the customer or an error message if it was not.</p> <p>Error messages may be generated:</p> <ul style="list-style-type: none"> A. In the RestFX validation layer: if the order is incorrectly formatted or does not comply with the business validation rules implemented as part of the RestFX validation logic. B. In Swedbank's FX trading platform: if the order is correctly formatted and in compliance with the business validation rules of the RestFX validation layer but not in compliance with the business validation rules of the trade execution system. <p>Errors falling in category A could be e.g. requests containing contradictory parameters, missing mandatory parameters. I.e. requests that do not comply with the API specification. Requests to trade outside of the service opening hours, to trade amounts above what is allowed on the service or to trade non-supported currencies or tenors (see endpoints <code>currencypairs</code> and <code>tenors</code>) would also fall under this category.</p> <p>Errors falling in category B could be e.g. requests to trade currencies for which the API Service User does not hold accounts with the bank or to trade with specified settlement on non-banking days such as weekends or other bank holidays.</p> <p>Format of error messages will differ between the two categories and are described in detail in the following section.</p>
orders/{orderID}	GET	Returns the data of a previously sent order (see orders POST above). The order is specified using an <i>orderID</i> parameter (see Section 6.3.2). Note: orders may be in different states and may or may not have been executed.
orders	GET	Get orders for a particular date. Returns the data of one or more orders (see orders POST above) having been sent on a specific date.

5.3 POST order – request and response specifications

Request – specification

The POST order endpoint takes a set of parameters specifying the exact details of the FX transaction the user wishes to conduct. If the request and the data provided is validated successfully in the RestFX validation layer an order object will be created in RestFX that can subsequently be passed on to “the market” i.e. to Swedbank’s FX trading platform.

Parameter	Definition	Format
<i>amount</i>	The nominal amount to be traded (exchanged). Expressed with exactly two decimals.	String
<i>amountCurrency</i>	The currency code for the currency in which <i>amount</i> was specified. Following ISO 4217 standard. (See also <i>side</i> below).	String
<i>currencyPair</i>	The currency pair to be traded. Made up of the codes for a <i>Base</i> and a <i>Quote (Price)</i> currency both following the ISO 4217 standard. E.g. EURSEK. Must be one of the currency pairs returned by the GET <i>currencypairs</i> endpoint.	String
<i>externalId</i>	The API Service Users own reference for this specific order ¹³ . This could contain the reference to a payment, invoice or similar to provide an audit trail for the whole transaction chain. See also <i>meansOfPayment</i> below. This field is not mandatory and is not validated by the bank. Maximum 50 characters.	String
<i>meansOfPayment</i>	Indicates the reason for the order. This field may only hold values “HEDGE” or “INVESTMENT”. HEDGE: the order is for the purpose of facilitating payment for identifiable goods or services (e.g. entering into an FX forward in order to pay an upcoming invoice in a foreign currency, or in preparation of an upcoming purchase in a foreign currency, as opposed to trading for investment purposes). INVESTMENT: any order made for investment purposes. Note: with tenors beyond Spot (“SP”) i.e. for all trades settling more than two banking days in the future the bank will only accept orders qualifying for the <i>Means Of Payment</i> exemption from MiFID II. This means that trades not having been classified as such using the “HEDGE” flag will be rejected by the bank.	String
<i>settlementDate</i>	An explicitly stated, expected, settlement date, for the trade. If the specified date is a non-Banking day the order will be rejected by Swedbank’s FX trading platform and an error message will be returned. Note: this parameter cannot be supplied in combination with the <i>tenor</i> parameter (the two are mutually exclusive).	String
<i>side</i>	Specifying whether the <i>Base</i> currency of the <i>currencyPair</i> is to be bought (“BUY”) or sold (“SELL”) against the <i>Price</i> currency.	String

¹³ Note: this is not to be confused with the *x-request-id* described earlier which is an identifier generic to the Open banking platform and used for all services published within that context but with no inherent business meaning. The intention with the *externalId* which is RestFX specific is to allow the API Service User to create the link to e.g. an invoice or similar.

	<p>E.g. using side “BUY” for the currencyPair “EURSEK” indicates that EUR is being bought against SEK. I.e. the customers EUR account will be credited, and the SEK account debited.</p> <p>This is not to be confused with the amountCurrency which is used to specify in which currency the nominal is defined.</p> <p>E.g. an order with:</p> <p style="text-align: center;"><i>side: “BUY”, currencyPair: “EURSEK”, amountCurrency: “SEK”, amount: “1000.00”</i></p> <p>would result in a transaction to buy EUR, with the customer being debited 1000 SEK and credited the corresponding EUR amount resulting from the exchange rate applied.</p> <p>The amount of EUR received in this case would be specified in the <i>counterAmount</i> field of the response (see Section 0).</p>	
<i>tenor</i>	<p>The tenor specifies the expected time to settlement defined as:</p> <p>TD (Today): same day settlement, available until 17:00 CET on banking days. TM (Tomorrow): settlement on the following banking day. SP (Spot): settlement two banking days in the future. 1W (One week): settlement one week after the SP date. 1M (One month): settlement one month after the SP date. 1Y (One year): settlement one year after the SP date.</p> <p>The tenor specified in the request must be one of the tenors listed by the <i>tenors</i> endpoint.</p> <p>If falling on a non-banking day the actual settlement date is normally rolled forward to the nearest following banking day. Note: this parameter cannot be supplied in combination with the <i>settlementDate</i> parameter (the two are mutually exclusive).</p>	String
<i>timeout</i>	<p>Instruction of how long the RestFX endpoint should wait for an order to be executed before cancelling the order and returning a time out response. I.e. this parameter specifies how long the requesting party is prepared to stand by for the response from the service. RestFX will either respond within this time, with the details of the trade having been executed or an error message, or it will respond with an error message once the timeout period has passed. If this happens RestFX will also attempt to cancel the order.</p> <p>Minimum value is 500 milliseconds i.e. half a second and maximum value is 20000 milliseconds i.e. 20 seconds.</p>	String

Request – Example, using tenor

Example of complete *order* object, using a tenor to specify the desired delivery date of the funds. In the RestFX data model this is made up of the Json body:

```
{
  "amount": "1000.00",
```

```

    "amountCurrency": "SEK",
    "currencyPair": "EURSEK",
    "externalId": "Invoice 12345",
    "meansOfPayment": "HEDGE",
    "side": "BUY",
    "tenor": "6M",
    "timeout": "11000"
  }

```

This is a request to buy ("side": "BUY") EUR in exchange for SEK ("currencyPair": "EURSEK"). The amount of SEK ("amountCurrency": "SEK") to be sold in this case is 1000 ("amount": "1000.00"). The number of EUR being delivered in return will depend on the FX rate for the specific trade resulting from this order (see Section 6.3.2). Note: It is possible to specify an exact EUR amount to be bought instead in which case the amountCurrency would have been EUR and the amount field would contain the amount in EUR that the customer wishes to procure.

The requested settlement date is on the first banking day 6 months after the Spot date ("tenor": "6M"). The exact date would be part of the response message.

In the request it has been specified that the order is for the purpose of facilitating payment for identifiable goods or services ("meansOfPayment": "HEDGE"). Had this not been the case the order would have been rejected as the tenor/settlement date is beyond Spot i.e. more than two banking days in the future.

The customer's own reference on this order is "Invoice 12345" ("externalId": "Invoice 12345"). This value is not validated by the bank in any way but kept as part of the logs to provide an audit trail for the API Service User.

The requesting party will listen for 11000 milliseconds ("timeout": "11000") for the response to the request before a cancellation of the order will be automatically sent.

Example – using a broken tenor (specifying a specific settlement date)

Same as the previous example but with a settlementDate specified instead of a tenor. Whether or not this order would be accepted by the bank and result in an executed trade depends whether the specified date, 2020-11-07, is a banking day for both EUR and SEK.

```

{
  "amount": "1000.00",
  "amountCurrency": "SEK",
  "currencyPair": "EURSEK",
  "externalId": "Invoice 12345",
  "meansOfPayment": "HEDGE",
  "settlementDate": "2020-11-07",
  "side": "BUY",
  "timeout": "11000"
}

```

Response - successfully executed trade

If the request has been properly validated in the RestFX validation layer an *order* object will be created and passed on “to the market” i.e. to Swedbank’s FX trading platform. The *trade Response* following a successfully executed trade will contain an *fxOrder* object that in its turn contains the entire dataset of the *order* object from the request with the additional data that defines the executed trade. Parameters part of the *order object* already covered in section 6.3.1 will not be described further here. Additional parameters, proprietary to the RestFX layer, are placed directly under the *tradeResponse* structure itself.

Parameter	Definition	Format
<i>tradeResponse</i>		
<i>orderID</i>	Order ID generated by RestFX. A unique identifier for the <i>order</i> object as created in RestFX once the request from the API Service User passed the RestFX validation layer. This ID is used when requesting the order from the <i>orders/{ID}</i> endpoint.	Numeric
<i>timestamp</i>	Timestamp in Coordinated Universal Time (UTC) EPOCH format specifying the time the order object was created in RestFX.	Numeric
<i>fxOrder</i>		
<i>externalID</i>	See section 6.3.1	String
<i>amount</i>	See section 6.3.1	String
<i>currency</i>	See amountCurrency of section 6.3.1	String
<i>currencyPair</i>	See section 6.3.1	String
<i>side</i>	See section 6.3.1	String
<i>message</i>	Optional. This field is only present when the FX trading platform returns an error.	
<i>tenor</i>	See section 6.3.1	String
<i>executionTime</i>	Timestamp for trade execution in the FX trading platform. In UTC format.	String
<i>executionRate</i>	FX rate at which the trade was executed. ($executionRate = spotRate + forwardPoints$).	Numeric
<i>counterAmount</i>	Traded (exchanged) amount in the currency not specified as the <i>amountCurrency</i> .	Numeric
<i>spotRate</i>	Spot rate of the trade. Same as <i>executionRate</i> for spot trades.	Numeric
<i>forwardPoints</i>	Forwards points ¹⁴ of the trade if applicable (Forward trades only). Expressed as “pips” i.e. as ten-thousandths and added or subtracted from <i>spotRate</i> to achieve the <i>executionRate</i> .	Numeric
<i>UTI</i>	Unique Trade Identifier (UTI) used in transaction reporting under EMIR. UTI:s are produced in the FX trading platform for all executed trades disregarding tenor/settlement date.	String
<i>fxOrderID</i>	A unique identifier for the trade executed. Generated by the FX trading platform. Note: this identifier is not the same as the <i>orderID</i> nor is it the same as the transaction ID that will be stated in the final trade confirmations send to the customer. It may however be used in communication with the bank should there be any questions related to settlement, reporting or for any other reason.	String
<i>settlementDate</i>	Date of delivery i.e. the date when actual settlement of the trade will take place. Note: if a <i>settlementDate</i> was specified in the original request (see section 6.3.1) this will be that same date. If a <i>tenor</i> (see section 6.3.1) was specified this will be the date as defined by the FX trading platform (and market convention) for that <i>tenor</i> .	String
<i>orderStatus</i>	The state of the initial <i>order</i> object as registered in RestFX. Possible states:	String

¹⁴ Forward points are the number of “pips” added to, or subtracted from, the spot rate of a currency pair to determine the forward rate for delivery on a specific settlement date ≠ the spot date. For trades settling on the Spot date the Forward points are always 0.

	<ul style="list-style-type: none"> • Created • Pending • Rejected • Failed • Cancelled • Booked • Unknown 	
<i>meansOfPayment</i>	See section 6.3.1.	

Response – example of successfully executed order

Example of response for a successfully executed order:

```
{
  "orderId": 2,
  "timestamp": 1588876583918,
  "fxOrder": {
    "externalId": "Refererens",
    "amount": "2.00",
    "currency": "EUR",
    "currencyPair": "EURSEK",
    "side": "SELL",
    "tenor": "SP",
    "executionTime": "2020-03-02T13:46:01.050 CET",
    "executionRate": 10.5955,
    "counterAmount": 21.19,
    "spotRate": 10.5955,
    "forwardPoints": 0,
    "UTI": "FX20200302EFXSPPRD.OS.1.1.735903",
    "fxOrderId": "1682090",
    "settlementDate": "2020-03-04"
  },
  "orderStatus": "Booked",
  "meansOfPayment": "HEDGE"
}
```

Response – Error generated in RestFX validation layer (A)

Errors generated in the RestFX validation layer as a result of an incorrectly formatted request or a request that is non-compliant with the business rules implemented on RestFX result in responses in the form of *tppMessages* structures with the parameters as listed below. In this case no order *object* has been created on the RestFX service (as the request failed validation and could not be parsed to a valid *order*) and the request was never passed on to the FX trading platform so there is no *fxOrder* either. I.e. the response is a short and hopefully precise description of the nature of the error and what the user needs to amend in the request in order to actually execute a trade.

Parameter	Definition	Format
<i>tppMessages</i>		
<i>code</i>	Error identifier.	String
<i>text</i>	Description of the root cause of the error.	String
<i>category</i>	Error.	String

Response – example of Error generated in RestFX validation layer (A)

Example of *tppMessages* structure:

```
{
  "tppMessages": [
    {
      "code": "A32",
      "text": "Service closed. Outside of opening hours.",
      "category": "ERROR"
    }
  ]
}
```

In this case the request was send outside of the opening hours defined in RestFX and the response consists of an error message stating as much.

Note: a request may be rejected by the RestFX validation layer for several reasons, the *tppMessages* response will however only contain one (1) error message at a time.

Response – Error generated in Swedbank’s FX trading platform (B)

If the request has been properly validated in the RestFX validation layer an *order* object will be created and passed on “to the market” i.e. to Swedbank’s FX trading platform.

If the *order* is rejected by the trading platform the response back will consist of a *tppMessages* structure (see section 6.3.3) containing an error and encapsulating a *tradeResponse* (see section 0) in turn encapsulating an *fxOrder* (see section 0).

Parameter	Definition	Format
<i>tppMessages</i>		
<i>tradeResponse</i>		
<i>fxOrder</i>		

This is also the format of the response for requests that time out due to e.g. network issues.

Response – example of Error generated in the back-end trading platform (B)

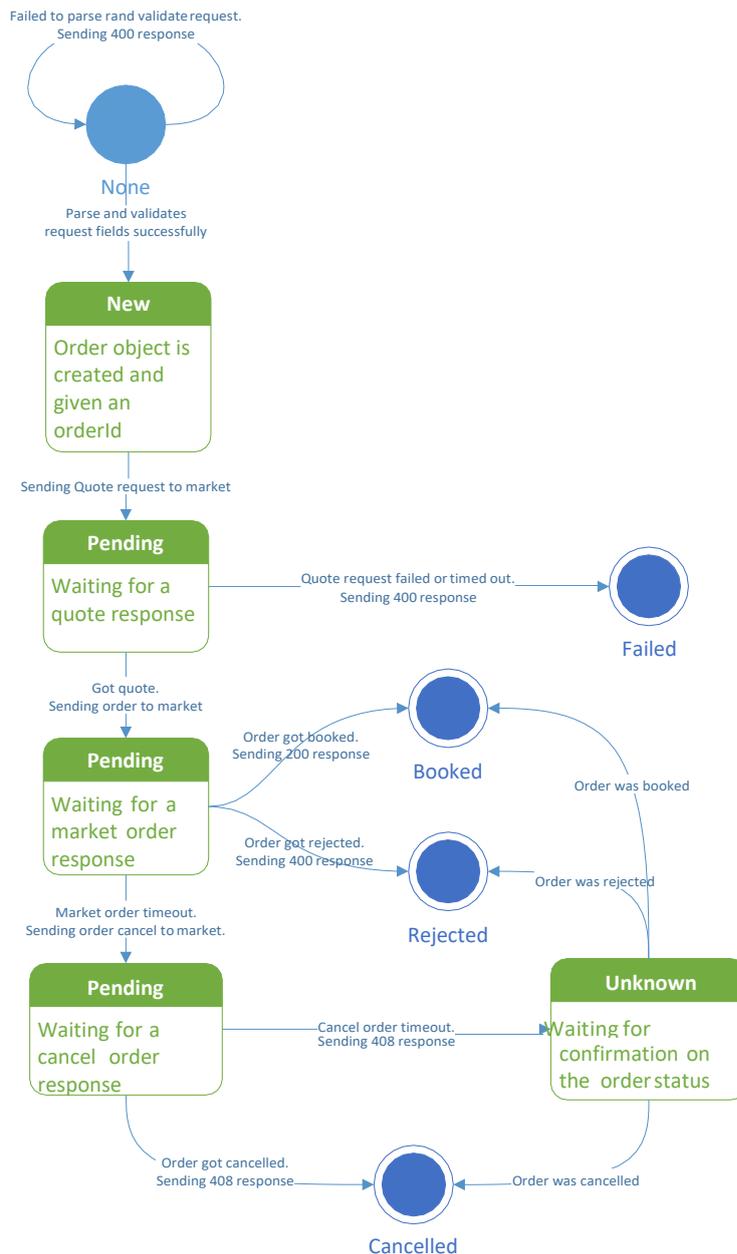
As no trade has been executed all parameters related to an actual trade execution (e.g. *executionRate* and *executionTime*) part of the *fxOrder* structure are null. The message parameter contains the error message returned by the FX trading platform.

```
{
  "tppMessages": [
    {
      "code": "A14",
      "text": "Temporary unavailable, please try again shortly",
      "category": "ERROR",
      "tradeResponse": {
        "orderId": 341,
        "timestamp": 1588920165147,
        "fxOrder": {
          "externalId": "123456abc",
          "amount": "1000.00",
          "currency": "EUR",
          "currencyPair": "EURSEK",
          "side": "BUY",
          "message": "Quote request was rejected by the market, reason Service currently unavailable,
                    with state: Retry",
          "executionTime": null,
          "executionRate": null,
          "counterAmount": null,
          "spotRate": null,
          "forwardPoints": null,
          "UTI": null,
          "fxOrderId": null
        },
        "orderStatus": "Failed",
        "meansOfPayment": "HEDGE"
      }
    }
  ]
}
```

5.4 Market order lifecycle and example usage

When an order is posted, the order will go through a series of steps in the backend. Depending on where the order is, it will be assigned different states. The state can be seen by looking at the `orderStatus` field in the `tradeResponse` object.

The following state diagram depicts the creation and lifecycle of an order:



Starting at **None**, the order object has not been created or given a state yet. It is first when the order request has been parsed and validated that an order object is created, stored and given a state. When the order has reached the state **New** it will be persisted in the system. From that point onwards it will be possible to request that order with a GET order request. When requesting an order, it is possible to get it in any of the above described states (though some are very unlikely).

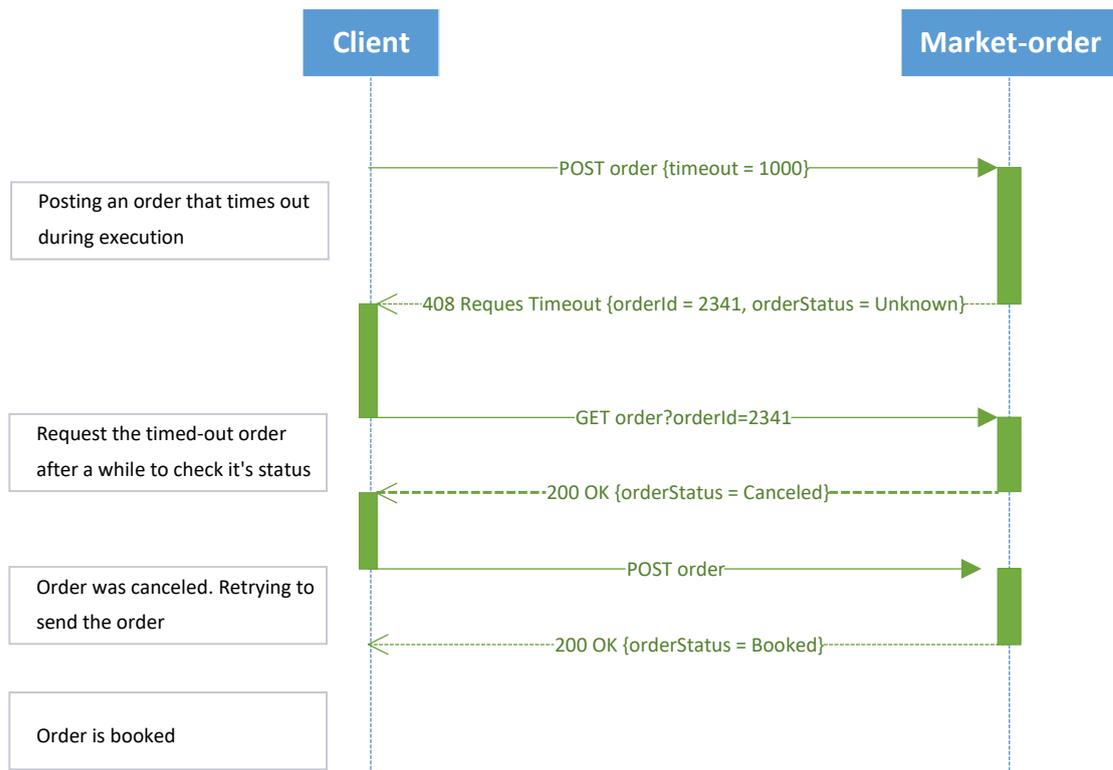
The below table describes the different states and suggested actions. The API Service User can always contact openbanking@swedbank.com¹⁵ to get help if the orders end up in unwanted states.

orderStatus	Description	Suggested Action
New	The order is persisted, given an orderId and populated with the fields provided in the request. This is a very short-lived state and will immediately change as soon as the order has been sent to the market.	Allow 5 seconds for processing in the back end before requesting an updated status of the order (GET order).
Pending	The pending status has a couple of sub states. The main feature of this state is that the backend is waiting for a response from the FX trading platform. Note: the order may continue from this state to any of the end states as defined below depending on availability of the back end FX trading platform and the subsequent validation of the order.	Wait until the backend process has completed the order. Allow 5 seconds for processing in the back end before requesting an updated status of the order (GET order).
Failed	This is an end state that signifies that it was not possible to request a quote from the FX trading platform. This can either be because a field in the request was not accepted by the FX trading platform or because of a timeout.	If it was not because of a timeout, check the order fields sent in the request and make sure nothing is wrong with them. If it was because of a timeout, wait for a bit and try again. The API Service User can also try and adjust the timeout field sent with the request to see if this help.
Rejected	This end state signifies that the order got rejected by the FX trading platform. This is usually because of some account configuration and not because of a technical issue.	The API Service User should contact openbanking@swedbank.com or the sales representative at the bank to get help with investigating why the orders are failing
Booked	This end state signifies that the order went through successfully.	
Cancelled	This end state happens when the timeout is reached.	Wait for a while and try again. The API Service User can also try to adjust the timeout field in the request.
Unknown	This state is reached when the timeout is reached, and RestFX fails to get a response from the cancellation of the order. This will happen if connection is lost to the FX trading platform at the same time as an order is in progress. Thus, it is a very unlikely state. When the connection to the FX trading platform is re-established, the state of the order should become known so long as the FX trading platform has not suffered data loss. In case of data loss, the bank will have to investigate and manually change the state.	Wait and see if the status changes. Allow 5 seconds for processing in the back end before requesting an updated status of the order (GET order). Contact openbanking@swedbank.com if the state of the order is not updated.

¹⁵ Please refer to <https://www.swedbank.com/openbanking/report-an-issue.html> for valuable advice on what information to provide when contacting the Open Banking support. This will speed up processing significantly.

For the most part, the usage of the API is simple. The API Service User sends an order and gets a trade response back with either a success or a failure.

It becomes more complicated when a timeout occurs or if a GET request is made for an order that is not yet in an end state. In that case, the API Service User must wait and request the order again after a while to check if the state has reached an end state. This sequence diagram describe an example of how this could be handled:



Under normal circumstances the response from RestFX is instantaneous and the order state included in the response is one of the end states (Failed, Rejected, Booked, Cancelled).

If the order still has not reached it's end state i.e. is in a state likely to change shortly (New, Pending, Unknown) the API Service User may need to explicitly request an update in the order status.

To avoid unnecessary load on the open banking and RestFX infrastructure the API Service User should allow 5 seconds for processing in the back end before requesting an updated status of the order (GET order).

6 RestFX API versioning guidelines

This section describes the guidelines for handling of changes and new releases of the RestFX API:s.

6.1 Versioning policy

- RestFX follows the principle of minimizing the number of versions. As a principle the API version is unaffected if changed or new functionality does not break backwards compatibility.
- The API Service User is recommended to keep the same version for all versioned RestFX API endpoints used.
- The aim is always to implement each change in the API:s without breaking backward compatibility. Changes that may be introduced without changing the version number include:
 - Adding of new functionality that does not directly impact the API specification
 - Adding new endpoints
 - Adding optional parameters (fields, headers, attributes) in requests and responses
 - Removal or loosening of constraint for input fields validated via server-side business logic
 - Adding additional error message(s)
- New version is created in case of:
 - Removal of endpoint or functionality
 - Adding mandatory parameters (fields, headers, attributes) in requests
 - Changes of data types or semantics of field
 - Tightening of constraints for input fields validated via server-side business logic
 - Removal of endpoint (or URI change);
- Only major versions is used in RestFX API URI endpoint. E.g.
<https://se.psd2.api.swedbank.com/partner/v1/fx/indicative-rate/currencypairs?app-id=..>

6.2 Documentation and information about changes in API versions

Breaking changes

Breaking changes, when a new version is introduced, are communicated three months in advance and API Service Users have an additional one month to adjust to the changes before the previous version of the API is decommissioned.

Timeline	Communication	Sandbox	Production
Release date - 3 months	Newsletter, Developer portal, Developer Documentation, Swagger	V[n] version available V[n+1] version available	-
Release date	Newsletter, Developer portal	V[n] version available V[n+1] version available	V[n] version available V[n+1] version available
Release date + 1 month	Newsletter, Developer portal	V[n+1] version available	V[n+1] version available

- Developer Documentation, technical documentation (Swagger) and Sandbox environment for new API versions are released at least 3 months before Production release. Information is provided in Developer Portal and in the newsletter distributed to all API Service Users.
- The bank is planning to support not more than 2 major API versions in production in parallel;
 - Parallel running is planned for 1 month
 - In special cases (if API Service User makes request and it is technically possible) parallel run can be extended up to 3 months

Non-breaking changes

With non-breaking changes the (major) version numbering in the URI remains unchanged.

Timeline	Communication	Sandbox	Production
Release date	Newsletter, Developer portal, Developer Documentation, Swagger	Non-breaking change V[n].x version available	Non-breaking change V[n].x version available

- When non-breaking changes are released, Developer Documentation, technical documentation (Swagger) and Sandbox environment for changes not requiring new major API version is released no later than Production date; Information is provided in Developer Portal and newsletter.