THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

openFinance API Framework

Operational Rules for Extended Services

Request to Pay Services

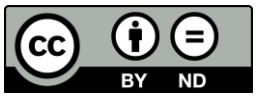
Version 2.0

20 October 2023

License Notice

This Specification has been prepared by the Participants of the openFinance Taskforce* (hereafter: Joint Initiative). This Specification is published by the Berlin Group under the following license conditions:

- "Creative Commons Attribution-NoDerivatives 4.0 International Public License"

  This means that the Specification can be copied and redistributed in any medium or format for any purpose, even commercially, and when shared, that appropriate credit must be given, a link to the license must be provided, and indicated if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. In addition, if you remix, transform, or build upon the Specification, you may not distribute the modified Specification.

- Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The Berlin Group or any contributor to the Specification is not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights**.**
- Any right, title and interest in and to the copyright and all related rights in topic-related Scheme Rulebooks, belong to the respective Scheme Manager (amongst others, the European Payments Council AISBL - EPC).
- The Specification, including technical data, may be subject to export or import regulations in different countries. Any user of the Specification agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import (parts of) the Specification.

---

* The openFinance Taskforce brings together participants of the Berlin Group with additional European banks (ASPSPs), banking associations, payment associations, payment schemes and interbank processors.

Contents

1 Introduction....1
1.1 From Core XS2A Interface to openFinance API....1
1.2 Request to Pay Services....2
1.3 Document Structure....4
2 RTP actors/roles and the mapping to the openFinance API....6
2.1 Basic model for RTP....6
2.2 Distinction between RTP and payment....8
2.3 Mapping of RTP to the openFinance API Framework for Receiver-RTPSP....9
2.4 Mapping of RTP to the openFinance API Framework for Originator-RTPSP-Aggregator....13
2.5 Mapping of RTP to the openFinance API Framework for Originator-RTPSP....14
2.6 Consent management....15
2.7 Future possible evolution of the RTP Services of the openFinance API Framework....17
3 Request to Pay Services of the openFinance API....19
3.1 Services of the openFinance API....19
3.2 Services covered in this document....19
3.2.1 Basic services for passive interface....20
3.2.2 Enhanced services for passive interface for the future....20
4 Actors and roles....22
4.1 Client related scenario....22
4.2 Direct access scenario....22
5 API Services supported for the Request to Pay Services....23
5.1 RTP basic services for passive interface....23
5.1.1 API Service: Establishment of RTP consent....24
5.1.2 API Service: RTP Initiation....26
5.1.3 API Service: RTP Cancellation....27
5.1.4 API Service: RTP Status....28
5.2 RTP enhanced services for passive interface....29

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

6 Example: Complete flow for RTP and corresponding payment 30
7 Key concepts of the openFinance API for the Request to Pay Service ... 32
7.1 Confirmation of the consent of the PSU (towards the API Client)... 32
8 Operational rules 33
8.1 API-Client authorisation 33
8.2 API-Client identification 33
8.3 Coding of business data 33
8.4 Payment initiation corresponding to an RTP 33
8.5 Payment initiation corresponding to an RTP via a PISP function ... 33
8.6 RTP Cancellation 34
8.7 Identification of RTP-Originator 34
8.8 Identification of the Originator-RTPSP 34
8.9 Status of an RTP transaction 34
8.10 Status Notification 37
9 Message and data model 38
9.1 Protocol Level 38
9.2 Authorisation Data Model 38
9.3 Request to Pay related data model 38
9.3.1 RTP Initiation Request 39
9.3.2 RTP Initiation Response 41
9.3.3 RTP Cancellation Request 42
9.3.4 RTP Cancellation Response 42
9.3.5 RTP GET Status Request 42
9.3.6 RTP GET Status Response 42
10 Annex 43
10.1 Glossary 43
10.2 References 44
10.3 List of figures 45
10.4 List of tables 46

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

1 Introduction

1.1 From Core XS2A Interface to openFinance API

With [PSD2] the European Union has published a directive on payment services in the internal market. Among others [PSD2] contains regulations on services to be operated by so called Third Party Payment Service Providers (TPP) on behalf of a Payment Service User (PSU). These services are

- Payment Initiation Service (PIS) to be operated by a Payment Initiation Service Provider (PISP) TPP as defined by article 66 of [PSD2],
- Account Information Service (AIS) to be operated by an Account Information Service Provider (AISP) TPP as defined by article 67 of [PSD2], and
- Confirmation on the Availability of Funds Service (FCS) to be used by a Payment Instrument Issuing Service Provider (PIISP) TPP as defined by article 65 of [PSD2].

To implement these services (subject to PSU consent) a TPP needs to access the account of the PSU. The account is managed by another PSP called the Account Servicing Payment Service Provider (ASPSP). To support the TPP in accessing the accounts managed by an ASPSP, each ASPSP has to provide an "access to account interface" (XS2A interface). Such an interface has been defined in the Berlin Group NextGenPSD2 XS2A Framework.

This XS2A Framework is now planned to be extended to extended services. This interface is addressed in the following as **openFinance API**. This openFinance API differs from the XS2A interface in several dimensions:

- The extended services might not rely anymore solely on PSD2.
- Other important regulatory frameworks which apply are e.g. GDPR.
- The openFinance API can address different types of **API Clients** as access clients, e.g. TPPs regulated by an NCA according to PSD2, or corporates not regulated by an NCA.
- The extended services might require contracts between the access client and the ASPSP.
- While the client identification at the openFinance API can still be based on eIDAS certificates, they do not need to be necessarily PSD2 compliant eIDAS certificates.
- The extended services might require e.g. the direct involvement of the access client's bank for KYC processes.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Note: The notions of API Client and ASPSP are used because of the technical standardisation perspective of the openFinance API. These terms are analogous to "asset broker" and "asset holder" resp. in the work of the ERPB on a SEPA API access scheme.

Note: In implementations, the API services of several ASPSPs might be provided on an aggregation platform. Such platforms will be addressed in the openFinance API Framework as "API provider".

The following account access methods are covered by this framework:

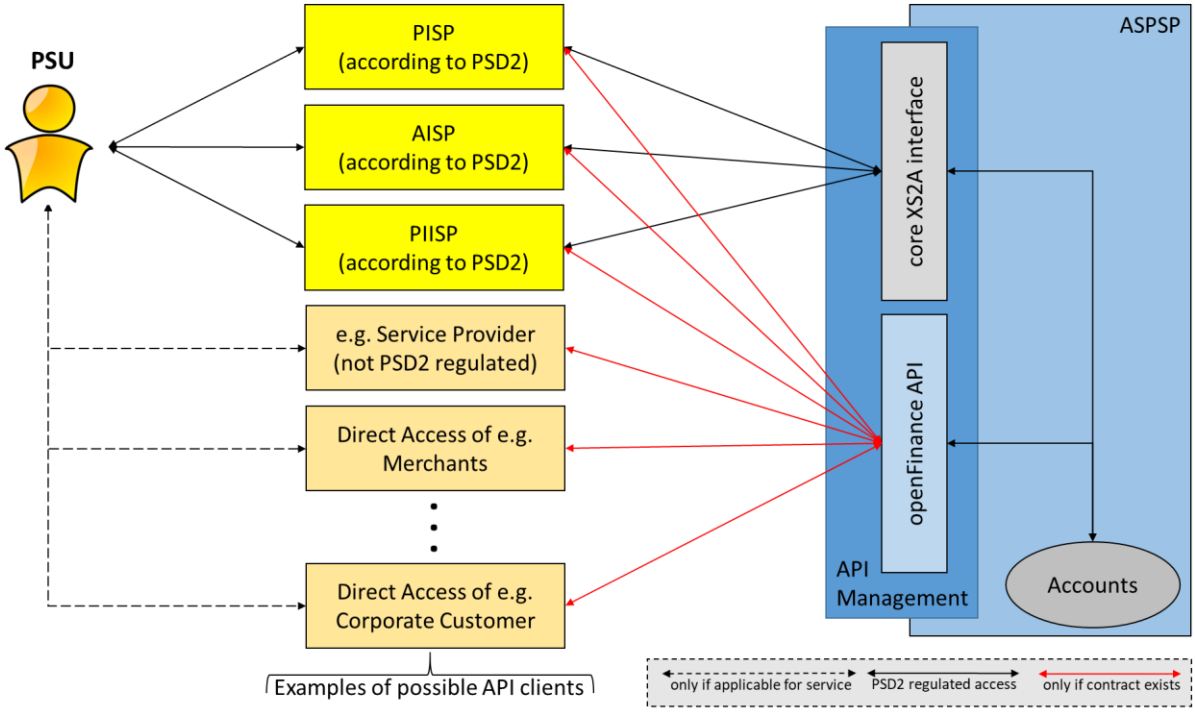


Figure 1: Core XS2A interface and openFinance API

The ASPSP may restrict the access to the services offered at its openFinance API and require dedicated onboarding. The requirements for the rights to access to services offered at the openFinance API are out of scope of this document. These requirements will be described in a dedicated operational rules document [oFA-OR-ADM].

1.2 Request to Pay Services

A Request to Pay scheme (RTP scheme) defines rules, messages and data sets for the exchange of a Request to Pay message (RTP message) from an originator to a receiver. Different RTP schemes in the sense of multilateral contractual systems or bilateral contracts are thinkable and already in the market or will come up in near future, but the basic architecture of these schemes will be very similar. Both, originator and receiver, are using Request to Pay service providers for the exchange of RTP messages, called Originator-RTPSP and Receiver-RTPSP. The communication between an Originator-RTPSP and a Receiver-RTPSP may be

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

done directly or it may be done using an intermediate Request to Pay service provider, called RTPSP-Aggregator.

With the Request to Pay Services of the openFinance API Framework, a service provider can use his openFinance API to receive RTP messages and to inform the sender about the status of the corresponding RTP transaction as well as potentially the status of the related payment. The major case considered in this document is, that an ASPSP will take over the role of a Receiver-RTPSP and will use his openFinance API to receive an RTP message, which is dedicated to one of his customers as receiver. Either an Originator-RTPSP or an RTPSP-Aggregator will take over the role of the API Client in this case. Please note that an abstraction to **non** ASPSP RTPSP-Receiver role is already feasible within this definition. In addition to this major case, the Request to Pay Service at openFinance APIs operated by an RTPSP Aggregator or even by an Originator RTPSP are described in addition. This is due to the fact that the related APIs are technically very similar to the major case mentioned above.

The Request to Pay Services of the openFinance API Framework are designed independently from actual RTP schemes. It is based on an abstract ISO20022 based model for Request to Pay as outlined in section 2.1. Technically it is based on a pure client/server model and the RESTful API approach, as it holds also for the overall openFinance API Framework. The smooth integration of the Request to Pay Services into the openFinance API Framework to allow for highest feasible synergies in implementations for the ASPSP as well as for the market demand side is one of the design principles for these Request to Pay Services. This integration allows then to benefit not only from technical basics like API client authentication routines but also from the openFinance payment data model and API access methods as well as existing push functionality

- either to the API client by pushing updated resource status information
- or from the Receiver-RTPSP (here ASPSP) to the receiver (here PSU) within the Push Account Information Service context.

The latter service might be of interest specifically for the case where the receiver is a corporate/SME.

In the positive case, a payment corresponds to an RTP message. For a general Request to Pay model the RTP message and the corresponding payment transaction are considered as two different processes. The Request to Pay Services considered in this Operational Rules document take for this an enhanced approach. Due to the integration into the openFinance API Framework, the Request to Pay Services are already enhanced by information about the status of corresponding payments. As soon as the receiver of the RTP message has initiated and authorised a corresponding payment, the API Client will be informed about this as part of the status of the RTP transaction. This is easily done at the end of the process chain since the PSU (as final receiver of the RTP message) will initiate the corresponding payment with his ASPSP, and this ASPSP is also the Receiver-RTPSP of the PSU receiving the RTP message from an API Client. In scenarios, where e.g. openFinance API server is the Originator-RTPSP, this still enables the Originator-RTPSP to forward the related information if received e.g. from the Receiver-RTPSP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

As a first step, only basic services to receive RTP messages and to inform about the status of an RTP transaction are supported by the Request to Pay Services of the openFinance API Framework. These basic services include already the information about the status of corresponding payments. For these basic services it is also possible that the RTP message may contain a payment related (electronic) document. For the basic services these documents will be forwarded to the receiver of the RTP as part of the presentment of the RTP to the receiver.

These basic services will be extended by future work on the openFinance API Framework. The enhancement of possible payment methods (like payment with a fixed payment schedule, payment with a loan) and the enhancement of specific rules for forwarding contained documents to the receiver of the RTP will be topics of this future work.

1.3 Document Structure

The document at hand gives an overview on the functional models of Request to Pay Services at an openFinance API as outlined in Section 3.2, after mapping the ISO20022 based RTP model to different openFinance API scenarios and roles of the RTP process chain in Section 2 Section 4 gives a short overview on third party based and direct access scenarios.

Section 5 then provides the description of the services supported by the first version of RTP services in the openFinance API Framework, resulting in a complete process flow overview in Section 6.

After having introduced consent considerations applying potentially to the RTP service in Section 7, the operational rules of the services are defined in Section 8. In the last Section 9, an abstract data model for API messages for RTP services is determined.

The technical specification of the Request to Pay Services in form of an API specification is not part of this document. Please refer to the corresponding Implementation Guidelines [oFA-IG-RtP] for this technical specification.

1.4 Document History

Version	Change/Note	Approved
0.9	Draft for public market consultation	2021-05-27
1.0	Added results from market consultation, i.e. • clarifications, • extending the API to the full process chain Originator – Originator-RTPSP Originator-RTPSP-Aggregator – Receiver-RTPSP • offering a bulk function for the interface Originator – Originator-RTPSP.	2021-09-24

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Version	Change/Note	Approved
	• Adapted the payment data model to the generic V2 payment data model of openFinance APIs	
2.0	Add document transmission and an optional RTP consent mechanism.	2023-10-20

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

2 RTP actors/roles and the mapping to the openFinance API

2.1 Basic model for RTP

Any Request to Pay (RTP) scheme will regulate and organise the cooperation of the following actors:

- **Originator** of the RTP. This can be for example an e-commerce shop. For the following payment corresponding to the RTP this actor will be the creditor.
- **Receiver** of the RTP. This can be for example a person or a company ordering something by the e-commerce shop. For the following payment corresponding to the RTP this actor will be the debtor.
- A service provider acting on behalf of the originator for forwarding an RTP. In this document this actor will be called **Originator-RTPSP**.
- A service provider acting on behalf of the receiver for receiving an RTP. In this document this actor will be called **Receiver-RTPSP**.
- In addition a special service provider can be involved on behalf of one or more Originator-RTPSP for executing the message exchange between an Originator-RTPSP and a Receiver-RTPSP. In this document this actor will be called **Originator-RTPSP-Aggregator**.

The following figure gives an overview of the actors and the message flow:

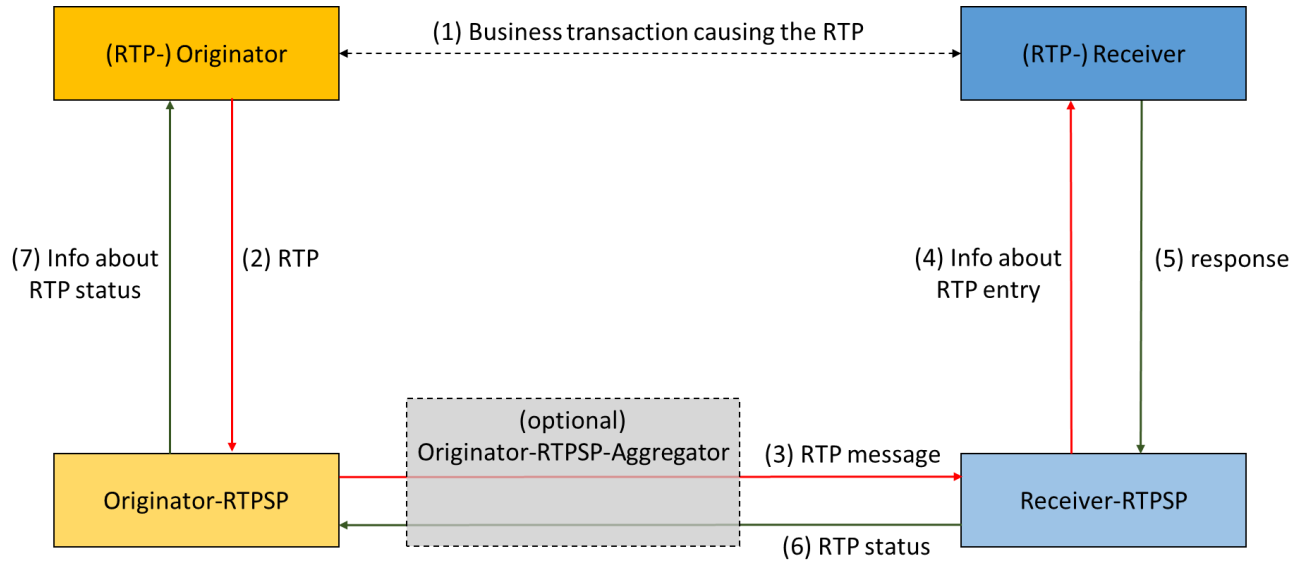


Figure 2: Basic model of Request to Pay

(ref. License Notice for full license conditions)

Remark: API services for step (4) are already covered by PUSH AIS services, cp. [oFA-OR-PAIS].

The information flow can be summarised as follows:

(1) Starting point is a business transaction between two entities, for example an order of some good by a person or a company at an e-commerce shop. Subsequently a Request to Pay message will be sent on behalf of the provider of the e-commerce shop (as RTP-Originator) to the person/company (as RTP-Receiver) who placed the order.

(2) For distributing RTP messages the RTP-Originator has mandated a service provider (as Originator-RTPSP). He informs this service provider about the RTP message to be sent and provides all necessary information for the RTP message, including potentially a related (electronic) document.

(3) The Originator-RTPSP builds the RTP message and sends this RTP message to the service provider, who has been mandated (as Receiver-RTPSP) by the person/company, who is the final receiver of this RTP message.

The Originator-RTPSP can send the RTP message either directly to the Receiver-RTPSP, if he has got direct access to this service provider, or he can send the RTP message to an Originator-RTPSP-Aggregator, who has been mandated with forwarding the RTP message to the correct Receiver-RTPSP.

(4) After receiving an RTP message the Receiver-RTPSP informs the final RTP-Receiver about the new entry of an RTP message. He provides the RTP-Receiver with all necessary information about the content of the RTP message, including the potentially included document.

(5) The RTP-Receiver decides about acceptance/rejection of the RTP and informs his service provider (Receiver-RTPSP) about his decision.

Remark:

Not only full acceptance/full rejection are possible at this point. The RTP-Receiver may decide to accept an RTP only partially or with changes. Following examples are possible beside others:

- An RTP may be accepted but the corresponding payment will be initiated earlier than the requested execution date of the RTP. This will be called "acceptance with changes" in this Operational Rules document.
- An RTP may be accepted but only with a reduced amount. This will be called "acceptance with changes" in this Operational Rules document.
- If the RTP message contains different sub-items, some sub-items may be accepted and other may be rejected. This will be called "partial acceptance" in this Operational Rules document. An example could be an RTP with an

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

machine readable e-invoice about five items included, but the RTP-Receiver accepts only the payment of three of the five items.

NOTE: The notion of dedicated acceptance of sub-items is not supported in this first version of the service and might follow at a later stage.

(6) The Receiver-RTPSP informs the service provider about the new status of the RTP message. This service provider can either be the Originator-RTPSP or the Originator-RTPSP-Aggregator, depending on who has sent the original RTP message.

The Originator-RTPSP or the Originator-RTPSP-Aggregator has got the possibility to get all necessary information about the response from the Receiver-RTPSP.

If an Originator-RTPSP-Aggregator has been informed about the new status of the RTP message, he will inform the Originator-RTPSP about this and will forward all necessary information about the new status to the Originator-RTPSP.

(7) The Originator-RTPSP informs the RTP-Originator about the new status and will provide all necessary information about the new status to the RTP-Originator.

2.2 Distinction between RTP and payment

Just for clarification: In this document Request to Pay is formally only seen as a system for exchanging RTP messages and not as a new payment system. Of course if the Receiver accepts an RTP, a corresponding payment transaction might be executed at once (depending on the payment authorisation status) or will be authorised/executed at a later time. For this payment the Receiver of the RTP will be the debtor and the Originator of the RTP will be the creditor. The processing of this payment is not part of the Request to Pay system, but will be done using known payment instruments like SCT or SCT Inst.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following figure shows this distinction:

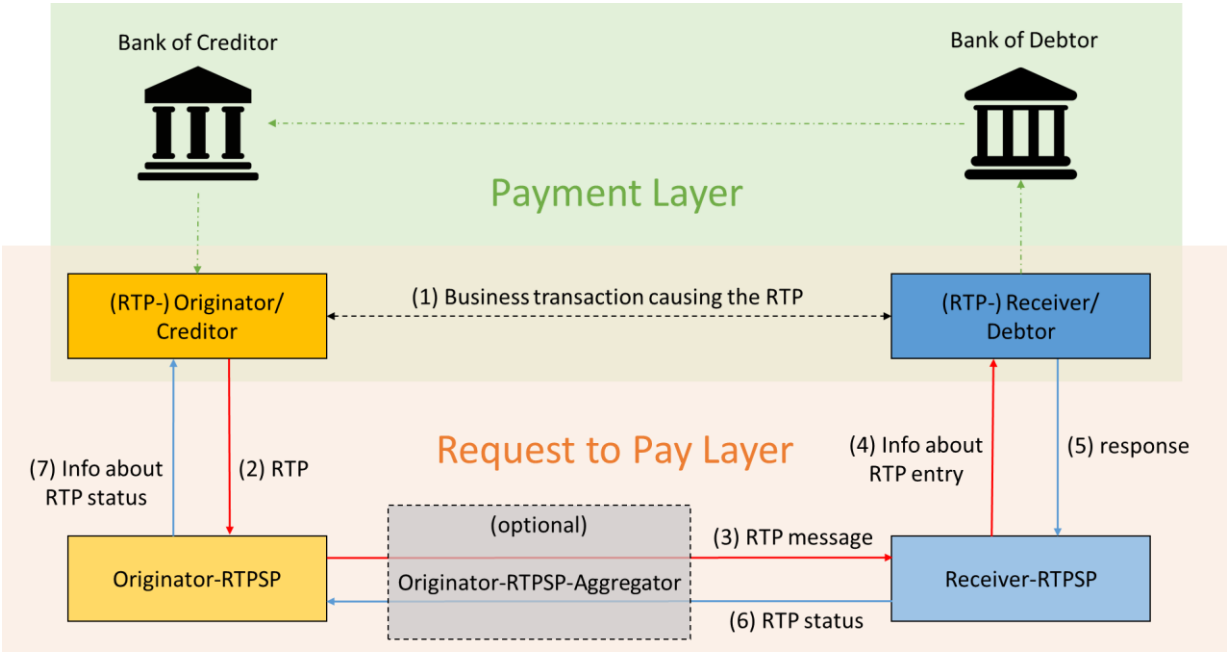


Figure 3: Distinction Payment Layer ↔ Request to Pay Layer

Due to this distinction the initiation and the execution of the corresponding payment within the payment layer are out of scope of the RTP Services of the openFinance API Framework, and hence out of scope of this Operational Rules document. Nevertheless, data for the initiation of the corresponding payment will be (in general) included in the RTP message and information about the status of the corresponding payment may be delivered as part of the RTP status message.

Remark:

- The next section will introduce the major case of RTP Services of the openFinance API Framework. In this case the bank of the debtor will also take the role of the Receiver-RTPSP. By this receiving an RTP message and initiating a corresponding payment can (technically) grow together even more.

2.3 Mapping of RTP to the openFinance API Framework for Receiver-RTPSP

The RTP Services of the openFinance API Framework will be developed in different evolutionary steps. See the next section 2.7 for more information and a perspective about this.

For the first step of the RTP Services of the openFinance API Framework the major case is considered, that the bank of the Receiver of an RTP (debtor bank, see section 2.2) takes over also the role of the Receiver-RTPSP for her customer. For the basic RTP Services the bank will support at her openFinance API services to receive RTP messages and to inform the

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

sender of the RTP messages about the status of the RTP message (and potentially also about the status of the corresponding payment (see section 8.9)).

The following table shows how the actors of the basic model for request to pay (Figure 2) are mapped to the roles of the openFinance API Framework:

Actor of Request to Pay (Figure 2)	Role of the openFinance API Framework
Receiver-RTPSP	ASPSP
(RTP-) Receiver	PSU
(RTP-) Originator	No role in the framework
Originator-RTPSP	API Client
Originator-RTPSP-Aggregator	

Table 1: Mapping of RTP actors to roles of the openFinance API Framework

REMARK: The assumption for this document that the Receiver-RTPSP is an ASPSP is used for guidance of ASPSPs which have implemented already other openFinance related ASPSP functionality. Still, these API definitions are not strictly requiring the Receiver-RTPSP to be an ASPSP and leave room for other institutions to offer this API as Receiver-RTPSP, e.g. Originator-RTPSP-Aggregator or insurance companies etc. as long as a related API access scheme allows this.

Please note that we have got for the basic RTP Services the following important difference from other services of the openFinance API Framework. While for other services the PSU participates via the API Client, for the RTP Services the PSU (as final receiver of the RTP) participates via the ASPSP.

While it is obvious for this first step of the RTP Services that the ASPSP (as the provider of the openFinance API) works as the Receiver-RTPSP and the PSU will be the RTP-Receiver, it is not so obvious what will be the task of the API Client. Will the API Client be the Originator-RTPSP or the Originator-RTPSP-Aggregator? To answer this question different scenarios have to be distinguished depending on the answers to the following questions:

- Who has done the necessary onboarding step as API Client at the openFinance API of the ASPSP according to [oFA-OR-ADM], the Originator-RTPSP or the Originator-RTPSP-Aggregator?
- Who's QWAC (qualified certificate for website authentication) is used to identify the client according to section 8.2, the QWAC of the Originator-RTPSP or the QWAC of the Originator-RTPSP-Aggregator?

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following table shows the scenarios considered for the cooperation of Originator-RTPSP and Originator-RTPSP-Aggregator if accessing an openFinance API for RTP:

Actor RTP	onboarding	QWAC	Role openFinance
Originator-RTPSP	X	X	API Client
	X	--	"indirect" API Client
	--	--	"transparent"
Originator-RTPSP-Aggregator	X	X	API Client
	--	X	"delegated" API Client
	--	--	"transparent"

Table 2: Scenarios considered for API Client

Not all of these theoretically possible combinations are useful. The roles "indirect" API Client and "delegated" API Client are not considered any longer. Only the following cases are considered as relevant:

Case a: The Originator-RTPSP takes over the role as API Client. He will do the onboarding at the openFinance API of the ASPSP for the RTP Services and the accesses to the API are secured by the QWAC of the Originator-RTPSP. If in this case the Originator-RTPSP will use the services of an Originator-RTPSP-Aggregator, the Originator-RTPSP-Aggregator is only a technical service provider and is transparent (not identified by QWAC) at the openFinance API of the ASPSP.

Case b: The Originator-RTPSP-Aggregator takes over the role as API Client. He will do the onboarding at the openFinance API of the ASPSP for the RTP Services and the accesses to the API are secured by the QWAC of the Originator-RTPSP-Aggregator. The Originator-RTPSP will be in this case transparent (not identified by QWAC) at the openFinance API of the ASPSP.

Based on this solution for the API Client we have got the following second important difference from other services of the openFinance API Framework. While for other services the technical service provider working as aggregators/intermediaries are always transparent (not identified by a QWAC) at the openFinance API, for the RTP Services service provider working as aggregators/intermediaries can take over the role of an API Client directly (case b).

Remarks:

- The API Client takes the liability for errors towards the ASPSP, regardless if he is the Originator-RTPSP (case a) or the Originator-RTPSP-Aggregator (case b).

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- The API Client has to pay any charges for using the RTP Services of the openFinance API of the ASPSP, regardless if he is the Originator-RTPSP (case a) or the Originator-RTPSP-Aggregator (case b).

- The initial RTP message received at the openFinance API will be based in its data model on the pain.013 message attributes of ISO 20022. But for synergy reasons with the payment services of the openFinance API this modelling will be integrated into the payment data model where payment initiation related and request related data will be easily distinguishable.

- Please note that subsequent RTP messages like retrieval of transaction status or cancellation are not based on pain.013, but are provided via usual generic RESTful API functionality for status retrieval or cancellation process provided within the openFinance API Framework.

- For case b: Even if the Originator-RTPSP is not identified at the openFinance API directly, he will be identified at application level by the content of the RTP message.

The following figure summarises the architecture for the first step of the basis RTP Services of the openFinance API Framework for the role where the Receiver-RTPSP is the API Server:

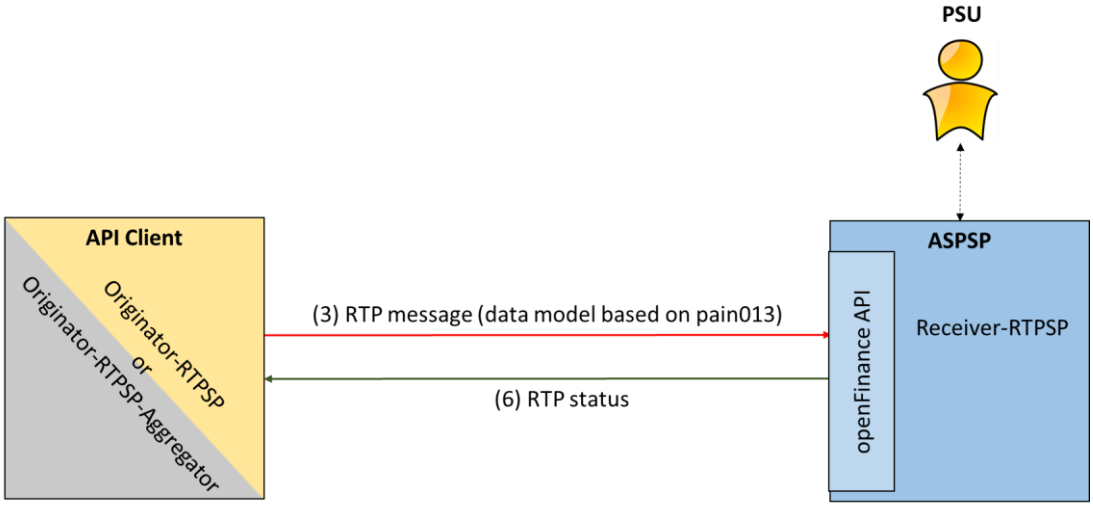


Figure 4: Architecture for the first step of the basic RTP Services of the openFinance API Framework

The openFinance API of the ASPSP supports API services which enables the ASPSP (in the role of the Receiver-RTPSP)

- to receive an RTP from an API Client (either the Originator-RTPSP or the Originator-RTPSP-Aggregator),

- to receive an RTS Cancellation request from an API Client (either the Originator-RTPSP or the Originator-RTPSP-Aggregator), and

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- to provide the corresponding RTP status to the API Client (either the Originator-RTPSP or the Originator-RTPSP-Aggregator).

The openFinance API is based on a pure API technology with the ASPSP as server and the API Client as client. The following two mechanisms can be used to inform the API Client about the status of an RTP:

- The API Client can do some polling for requesting the status of the resource which has been created when the RTP has been delivered, using a GET on the resource. The RTP status will be provided as part of the response data of this GET. This variant is always available due to the Restful API approach.

- The Resource Status Notification Function supported by [oFA-IG-ProtSec] will be used by the ASPSP to inform the API Client about any change of the status of the resource, possibly including some parameters of the new status. The subscription to this service is a generic openFinance API Framework function and will be "on the fly" with every first API access of the API Client for the Request to Pay transaction. If necessary the API Client can then use a GET on the resource, if he needs further information about the RTP. The information about the RTP will be provided as part of the response data of this GET method.

Remarks:

- Today, the Resource Status Notification function of an openFinance API only supports pushing information to the domain defined in the QWAC of the API Client due to security reasons.

- The status information presented by the ASPSP actively to the API Client (as part of the Resource Status Notification function) will inform about any status changes which has been applied to the RTP (resource). The status information will indicate if it is necessary to execute a GET on the resource to get further information or if all relevant new information is already contained in the status information.

- The content of the status information presented by the ASPSP as part of the Resource Status Notification function will be defined by the corresponding implementation guidelines [oFA-IG-RtP].

The first alternative shall only be used by an API Client if the Resource Status Notification function is not useable by the API Client. An ASPSP supporting the Request to Pay Services at his openFinance API for this interface shall support the Resource Status Notification function.

2.4 Mapping of RTP to the openFinance API Framework for Originator-RTPSP-Aggregator

In addition to the above, the case is considered, that the Originator-RTPSP-Aggregator is offering the openFinance RTP API to the Originator-RTPSP as an API Client. The Originator-

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

RTPSP-Aggregator receives RTP requests from the connected Originator-RTPSP and forwards these requests to the relevant Receiver-RTPSP. For this behalf, the related RTP message needs to contain routing information identifying the related Receiver-RTPSP.

The following table shows how the actors of the basic model for request to pay (Figure 2) are mapped to the roles of the openFinance API Framework in this scenario:

Actor of Request to Pay (Figure 2)	**Role of the openFinance API Framework**
(RTP-) Receiver	No role in the framework for this scenario
(RTP-) Originator	No role in the framework for this scenario
Originator-RTPSP	API Client
Originator-RTPSP-Aggregator	API Server

Table 3: Mapping of RTP actors to roles of the openFinance API Framework

Remark: In this scenario, the Originator-RTPSP-Aggregator as API Server is most likely not an ASPSP as such. But as said above, the notion to be an ASPSP is not necessary for the openFinance API Framework for Request to Pay Services.

The openFinance API of the Originator-RTPSP-Aggregator supports API services which enables the Originator-RTPSP-Aggregaror

- to receive an RTP from an API Client (the Originator-RTPSP),
- to receive an RTS Cancellation request from an API Client (the Originator-RTPSP), and
- to provide the corresponding RTP status to the API Client (the Originator-RTPSP).

2.5 Mapping of RTP to the openFinance API Framework for Originator-RTPSP

In addition to the above, the case is considered, that an ASPSP as an Originator-RTPSP is offering the openFinance RTP API to the RTP-Originator as an API Client. The Originator-RTPSP receives RTP requests from the connected Originator and

- Either has at the same time the role of the Receiver-RTPSP: This case is no longer described since the status information towards the API Client will be content wise analogous to the case where the ASPSP is the Receiver-RTPSP.
- Or will forward these requests to the relevant Receiver-RTPSP. For this behalf, the related RTP message needs to contain routing information identifying the related Receiver-RTPSP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following table shows how the actors of the basic model for request to pay (Figure 2) are mapped to the roles of the openFinance API Framework in this scenario:

Actor of Request to Pay (Figure 2)	Role of the openFinance API Framework
(RTP-) Receiver	No role in the framework for this scenario
(RTP-) Originator	API Client
Originator-RTPSP	ASPSP

Table 4: Mapping of RTP actors to roles of the openFinance API Framework

Remark: In this scenario, the Originator-RTPSP-Aggregator as API Server might not be an ASPSP as such. But as said above, the notion to be an ASPSP is not necessary for the openFinance API Framework for Request-to-Pay Services.

The openFinance API of the Originator-RTPSP supports API services which enables the Originator-RTPSP

- to receive an RTP from an API Client (the Originator),
- to receive an RTS Cancellation request from an API Client (the Originator), and
- to provide the corresponding RTP status to the API Client (the Originator).

Note: Please note that for this scenario only, the RTP message can be submitted either as single RTP transactions or as an RTP bulk message. The latter variant is optional for the RTPSP-Originator to be offered. The related detailed status information provided for RTPs submitted as RTP bulk are provided on single bulk entry only since in difference to payments, there is no further bulk processing (like booking procedures) on the ASPSP side as it is in payment initiations.

2.6 Consent management

An RTP should only be sent to an RTP-Receiver, if the RTP-Receiver has given his consent to receive RTPs to the (RTP-) Originator. In contrast to consents for account assess, this

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

standard is not mandating any requirement on the form of the related consent (could be a verbal consent, click consent, SCA usage etc).

Nevertheless, a consent might still be managed via the openFinance API. For this different types of consents can be distinguished:

RTP-Originator Consent:

> Consent given by the RTP-Receiver to an Originator-RTPSP to send RTPs of a given RTP-Originator to his Receiver-RTPSP.

Originator-RTPSP Consent:

> Consent given by the RTP-Receiver to an Originator-RTPSP to send RTPs of any RTP-Originator to his RTP-Receiver-RTPSP.

Trusted Originator List:

> Consent given by the RTP-Receiver to an RTP-Originator to send RTPs to his Receiver-RTPSP regardless of the Originator-RTPSP used.

PSU enrolment:

> Consent given by the RTP-Receiver to his Receiver-RTPSP with or without further restrictions about RTP-Originator or Originator-RTPSP

NOTE: A consent of type "PSU enrolment" has to be managed internally/bilateral between the RTP-Receiver and the Receiver-RTPSP. This is not managed at the openFinance API.

NOTE: The establishment of consents of type "Trusted Originator List" is not supported by the openFinance API of the ASPSP (as Receiver-RTPSP) for now, but may be supported in a later version of the standard.

The establishment of consents of type "RTP-Originator Consent", and "Originator-RTPSP Consent" will be supported by the openFinance API of the ASPSP (as Receiver-RTPSP). This has to be confirmed by the PSU (as RTP-Receiver) by executing an SCA. This will be implemented as extension to the Consent API [oFA-IG-Consent].

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Remarks:

- Consent management at the openFinance API is always an optional feature depending on the decision of the ASPSP.
- An ASPSP as Receiver-RTPSP may implement also internal processes to manage the consent or trust the Originator-RTPSP.
- It is a case by case decision of the ASPSP as Receiver-RTPSP if a consent is requested for the current transaction depending on fraud detection and risk management policies of the ASPSP.

2.7 Future possible evolution of the RTP Services of the openFinance API Framework

As already mentioned the Request to Pay Services of the openFinance API Framework will be developed in different evolutionary steps. The following gives an overview of the building blocks (functionality and restrictions) to be implemented during the evolution of the RTP Services of the openFinance API Framework.

Building block "basic"

All scenarios as introduced above can be supported.

The RTP message can contain an RTP or an RTP Cancellation request.

The RTP can contain the information of a pain013 ISO message.

The RTP can contain in addition to the payment information an electronic document, which is related to the payment. This document can serve for example as an electronic invoice. Special restrictions for formatting these documents may be introduced by the Receiver-RTPSP. This document will be forwarded to the PSU as part of the RTP presentment.

The RTP status can only distinguish between acceptance/rejection of an RTP. Partial acceptance will not be supported as part of this building block, But acceptance with changes (reduced amount for payment, earlier initiation of payment) will be possible.

Optional feature depending on the ASPSP: The consent of type "RTP-Originator Consent" given by the RTP-Receiver to an API Client to send RTPs for a given RTP-Originator can be established at the openFinance API.

Optional feature depending on the ASPSP: The consent of type "Originator-RTPSP Consent" given by the RTP-Receiver to an API Client to send RTPs without restrictions concerning the RTP-Originator can be established at the openFinance API.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Building block "enhanced"

The RTP message may contain a link to a document (instead of the electronic document itself).

Forwarding contained documents can be restricted to the occurrence of special events, for example the initiation of a corresponding payment.

The RTP status can distinguish also partial acceptance.

The RTP status can contain more information, for example the confirmation of a corresponding payment.

The consent of type "Trusted Originator List" given by the RTP-Receiver to an RTP-Originator to send RTPs regardless of the Originator-RTPSP can be established at the openFinance API.

See section 3.2 for further details.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3 Request to Pay Services of the openFinance API

3.1 Services of the openFinance API

The openFinance API supports different services. It might also subsume core services as they are offered in the core XS2A interface to fulfil the PSD2 requirements. In difference to the PSD2 compliant XS2A interface, the openFinance API is not necessarily requiring the API Client to be a TPP regulated by PSD2.

The ASPSP might mandate a contract between the API Client and the ASPSP for accessing the openFinance API. All potentially used administrative processes in the openFinance API, e.g. for onboarding processes are described in [oFA-OR-ADM].

Please note that for Request to Pay Services the notion of an ASPSP as API Server is not mandated from a PSD2 regulatory point of view, since formally speaking a Request to Pay is not a payment initiation. Nevertheless, the notion of an ASPSP is used for the API Server throughout this document, since this is used in all other documents of the openFinance API Framework cited in this document.

This openFinance API standard is not defining whether a service is to be supported in the openFinance API or not, but it might define a mandatory support of sub-services once a dedicated openFinance API service is offered.

3.2 Services covered in this document

The following **sub services** of Request to Pay Services are addressed in this document within the openFinance API Framework:

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3.2.1 Basic services for passive interface

Services addressed	Usage
RTP basic services	Within this sub service, the ASPSP offers technically separated building blocks, by which he can • receive an RTP from an API Client, • receive an RTS Cancellation request from an API Client, and • provide the corresponding RTP status to API Client. The RTP may contain in addition to the payment information an electronic document related to the payment. Special restrictions for formatting this document may be introduced. This document will be forwarded to the PSU as part of the RTP presentment. For the status report the positive response of the debtor (i.e. the acceptance of the RTP) and the negative response (i.e. the rejection of the full RTP) are supported. Positive response may contain information about changes to the accepted RTP (reduced amount for payment, earlier initiation of payment). No partial acceptance/rejection for dedicated item lists is supported by the first step of the service.
RTP with payment confirmation	After the PSU has initiated the corresponding payment a new enhanced RTP status is provided to the API client containing a confirmation of the payment.
RTP consent Originator-RTPSP	Establishment of a consent of type "RTP-Originator Consent" or type "Originator-RTPSP Consent" given by the PSU to an Originator-RTPSP for sending RTPs with or without restrictions for a special RTP-Originator

Table 5: Request to Pay Services scheduled for step 1 of further detailed work

3.2.2 Enhanced services for passive interface for the future

The basic Request to Pay Services will be enhanced by further services, which are not covered by the first version of the Request to Pay services of the openFinance API Framework.

Services addressed	Usage
RTP enhanced by a link to a document	The RTP may contain a link to an electronic document (instead of the electronic document itself). No special requirements for formatting the

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Services addressed	Usage
	document are given. This document/link is forwarded to the PSU as part of the presentment of the RTP. No enhancement for the RTP status.
RTP enhanced by an e-invoice with partial acceptance/rejection	The RTP may contain an e-invoice or a link to an e-invoice. Special requirements for formatting the e-invoice are given which enables further processing of the e-invoice by the ASPSP. This e-invoice/link is forwarded to the PSU as part of the presentment of the RTP. The PSU may accept the e-invoice only partially. The RTP status is enhanced containing information about the partial acceptance/rejection of different positions of the e-invoice. The API Client may be informed about the reasons why some items will not be paid.
RTP with document or link to a document to be forwarded after payment.	RTP with or without an electronic document or link to a document. This document /link is not forwarded to the PSU as part of the presentment of the RTP. RTP status as above. After the PSU has initiated the corresponding payment the document/e-invoice/link is forwarded to the PSU. A new enhanced RTP status is provided to API Client containing a confirmation of the payment and of the forwarding of the document/link.
RTP with a pre-defined payment schedule	RTP with or without an electronic document as above. The RTP contains in addition a predefined payment schedule. RTP status as above. A new enhanced RTP status is provided after initiation of each corresponding payment containing a confirmation of the payment.
RTP consent RTP-Originator	Establishment of a consent of type "Trusted Originator List" given by the PSU to an RTP-Originator for sending RTPs without restrictions for a special API Client to be used.

Table 6: Request to Pay Services scheduled for step 2 of further detailed work

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

4 Actors and roles

4.1 Client related scenario

In general, services offered by an ASPSP at his openFinance API may be accessed/used not only by API Clients registered by an NCA in the role of a TPP according to the PSD2 regulation. Instead further service provider may take over the role of an API Client (see [oFA-OR-ADM]). This is also true for the Request to Pay Services.

4.2 Direct access scenario

The openFinance API Framework plans to develop best practices for re-using the API Client/ASPSP openFinance API also as an PSU-ASPSP interface, especially for the corporate case, where the broad functionality like multi-signing etc. applies. For the services at hand, this might be applicable specifically for the interface RTP-Originator and Ortiginator-RTPSP.

NOTE: Please note that direct access in this case is technically easy since no authorisation process i.e. no SCA is involved in the API as such.

For the Request to Pay Services the direct access of a PSU to his ASPSP as his Receiver-RTPSP is already defined by the PUSH AIS services.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

5 API Services supported for the Request to Pay Services

The current version of the Request to Pay Services supports the following API services. The **Service Type** is indicating a bundling of several API services to a service family.

5.1 RTP basic services for passive interface

For the first step the following service types are introduced:

- RTP basic services (for passive interface) (XRTP-B).
- Establishment of a consent (CONS-RTP).

The following table gives an overview of this API services:

API Services	Service Type	Technical Functionalities	PSU directly involved
RTP Initiation	XRTP-B	Delivery of an RTP	For these services not relevant.
RTP Cancellation	XRTP-B	Cancellation request	
RTP and payment status	XRTP-B	Request of an RTP status (including information about the status of the corresponding payment)	
Establishment of RTP consent	CONS-RTP	Establishment of a consent of type "RTP-Originator Consent" or type "Originator-RTPSP Consent"	Yes, SCA necessary

Table 7: API services of the RTP basic services for passive interface

In addition, the openFinance API will support technical use cases within the RESTful API approach which are not necessarily used within the above mentioned use cases, e.g. to read details on created resources.

Remark:

> Due to the integration of the Request to Pay Service into the architecture of the openFinance API (including the payment services), the status of an RTP might contain also information about the status of the corresponding payment already for the RTP basic services.

In the following sub sections the top level of the information flow for this single services is shown. Please see section 6 for an example of a combined flow of RTP and the initiation of the corresponding payment (including the SCA of the PSU).

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

It is assumed that these RTP basic services for a passive interface will be (almost) identical API calls regardless

- if the passive interface is operated by an ASPSP to receive RTP messages from an API Client (Originator-RTPSP or Originator-RTPSP-Aggregator),
- if the passive interface is operated by an Originator-RTPSP-Aggregator to receive RTP messages from an Originator-RTPSP, or
- if the passive interface is operated by an Originator-RTPSP to receive RTP messages from an RTP-Originator.

Details about this have to be checked as part of the specification of these basic services.

5.1.1 API Service: Establishment of RTP consent

This API service can be used by an API Client (Originator-RTPSP or Originator-RTPSP-Aggregator) to establish a consent given by a PSU (as RTP-Receiver) to an Originator-RTPSP. This consent enables the Originator-RTPSP to send an RTP to the PSU (as RTP-Receiver) using the openFinance API of the ASPSP (as Receiver-RTPSP for the PSU).

Remarks:

- For an ASPSP it is an option to support this API service. It is also possible to use internal ASPSP specific processes to establish the necessary consent or to trust the Originator-RTPSP in managing the consent between PSU and RTP-Originator.
- The consent is always given by the PSU to the Originator-RTPSP, regardless if the Originator-RTPSP itself accesses the openFinance API of the ASPSP as API Client or if an Originator-RTPSP-Aggregator accesses the openFinance API as API Client on behalf of the Originator-RTPSP.
- Only consents of type "RTP-Originator Consent" and type "Originator-RTPSP Consent" can be established by this process.
- For a consent of type "RTP-Originator Consent" the RTP-Originator and the Originator-RTPSP have to be identified correctly (see sections 8.7 and 8.8).
- For a consent of type "Originator-RTPSP Consent" the Originator-RTPSP has to be identified correctly (see section 8.8).

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following figure shows only the top level information flow for this API Service:

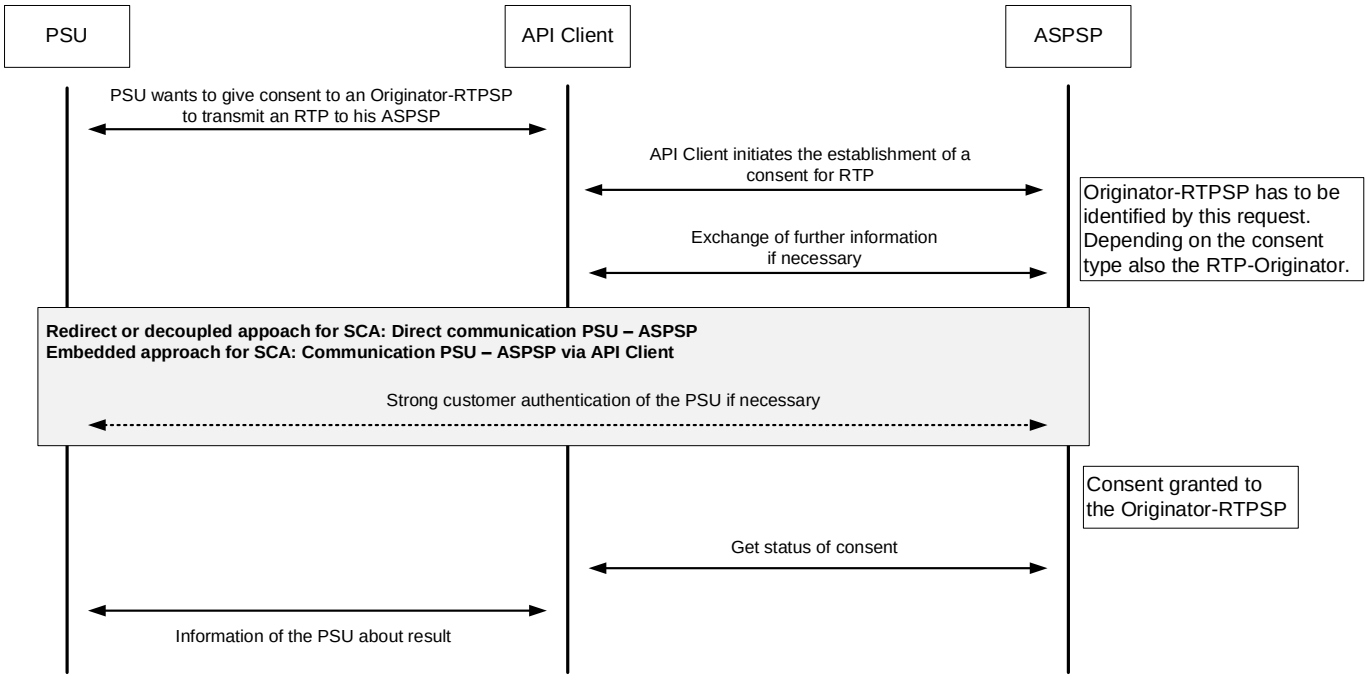


Figure 5: API Service Establishment of RTP consent

Details of the service will be specified as part of the consent API for version 2 of the openFinance framework [oFA-IG-Consent].

Note: Due to the optionality of the consent establishment via the openFinance API, the related consent resource identification is **not** used in a follow-up RTP initiation as a sort of access token, see below. This is in contrast to consent mechanisms for account information access.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

5.1.2 API Service: RTP Initiation

This API service can be used by an API Client (Originator-RTPSP or Originator-RTPSP-Aggregator) to deliver an RTP (based on an pain013 ISO-message and optionally containing an electronic document) to the ASPSP (as Receiver-RTPSP):

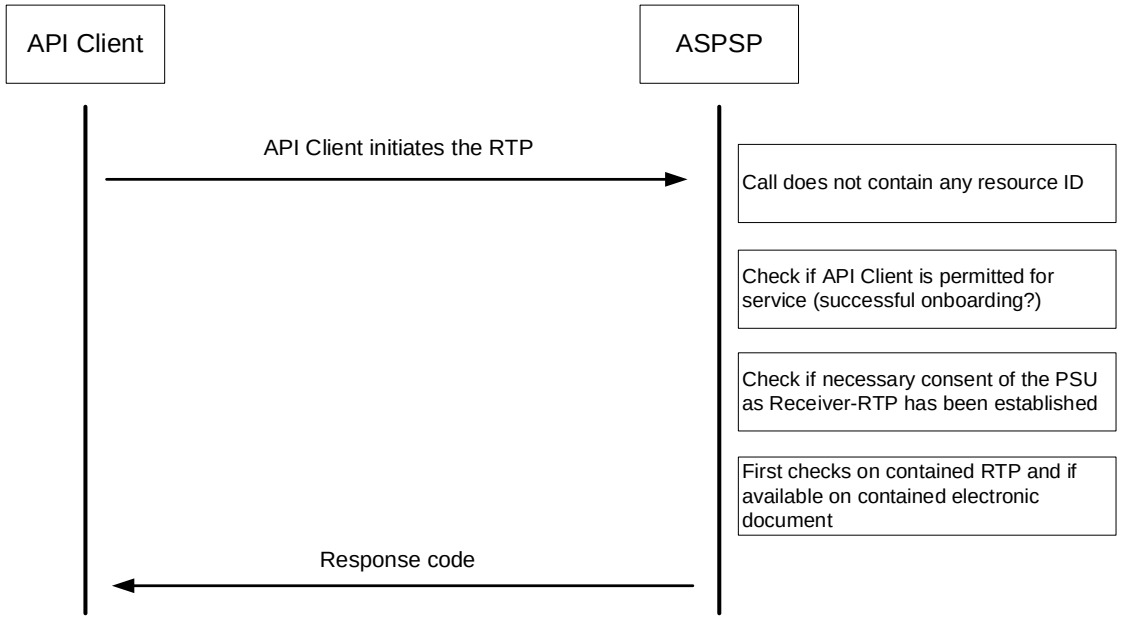


Figure 6: API Service RTP Initiation

Precondition:

The consent of the PSU to receive an RTP from the RTP-Originator sent by the Originator-RTPSP (consent type "RTP-Originator Consent") or to receive an RTP from the Originator-RTPSP regardless of the RTP-Originator (consent type "Originator-RTPSP Consent") has been established before. This is not shown in the figure above. This consent may have been established using the openFinance API of the ASPSP (as shown in section 5.1.1) or by internal processes specific for the ASPSP or just between the PSU and RTP-Originator/Originator-RTPSP.

Remark:

1) As already indicated before this API service may also be used

- by an Originator-RTPSP to deliver an RTP to an Originator-RTPSP-Aggregator, if the Aggregator supports the openFinance API with this service as its passive interface to its clients, or

- by an RTP-Originator to deliver an RTP to an Originator-RTPSP, if the Originator-RTPSP supports the openFinance API with this service as its passive interface to its clients. In this scenario, the detailed specifications will also allow to submit the

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

RTPs as bulk. Status information can then be provided either on bulk or on single entry level. Anyhow, cancellation processes will **only** be supported on single entry level. Details about this will be defined in the related implementation guidelines [oFA-IG-RtP].

2) For this "RTP Initiation" towards the Receiver-RTPSP as ASPSP, the API Client can be either the Originator-RTPSP or an Originator-RTPSP-Aggregator. In any case the Originator-RTPSP has to be identified correctly as part of the RTP Initiation (see also section 8.8):

- If the Originator-RTPSP is accessing the openFinance API of the ASPSP directly as the API Client, its identification is done by his QWAC used to access the API.
- If an Originator-RTPSP-Aggregator is accessing the openFinance API of the ASPSP directly as the API Client (on behalf of the Originator-RTPSP), it uses its own QWAC to access the API, hence the Originator-RTPSP cannot be identified by this QWAC. For this reason the request related data has to contain sufficient data to identify the Originator-RTPSP (see section 9.3.1.3).

5.1.3 API Service: RTP Cancellation

This API service can be used by an API Client (Originator-RTPSP or Originator-RTPSP-Aggregator) to request the ASPSP (as Receiver-RTPSP) to cancel an RTP delivered before:

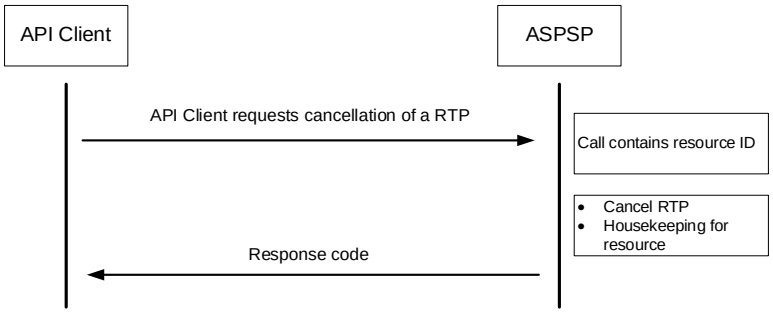


Figure 7: API Service RTP Cancellation

Remark:

As already indicated before this API service may also be used

- by an Originator-RTPSP to request an Originator-RTPSP-Aggregator to cancel an RTP delivered before, if the Aggregator supports the openFinance API with this service as its passive interface to its clients, or
- by an RTP-Originator to request an Originator-RTPSP to cancel an RTP delivered before, if the Originator-RTPSP supports the openFinance API with this service as its passive interface to its clients.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

5.1.4 API Service: RTP Status

This API service can be used by an API Client (Originator-RTPSP or Originator-RTPSP-Aggregator) to request from the ASPSP (as Receiver-RTPSP) the current status of an RTP delivered before. Depending on the approach for this two cases have to be distinguished for this API service.

Remarks:

- An ASPSP shall always support the Resource Status Notification Service at its openFinance API.
- The polling shall only be used by an API Client if the Resource Status Notification Service is not useable by the API Client.

Request the RTP status by polling the API of the ASPSP:

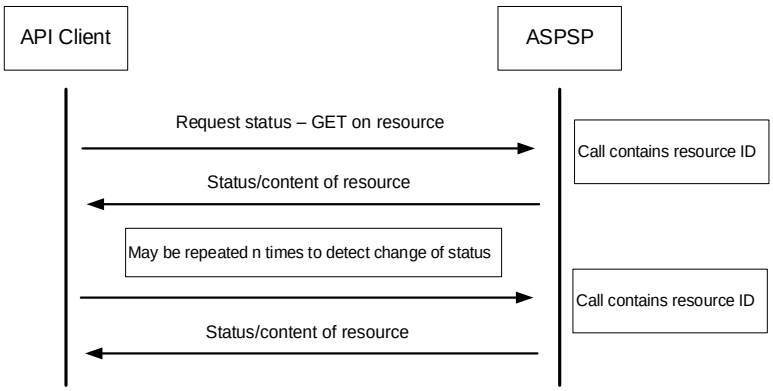


Figure 8: API Service RTP Status with polling

Request the RTP status by using the Resource Status Notification function of [oFA-IG-ProtSec]:

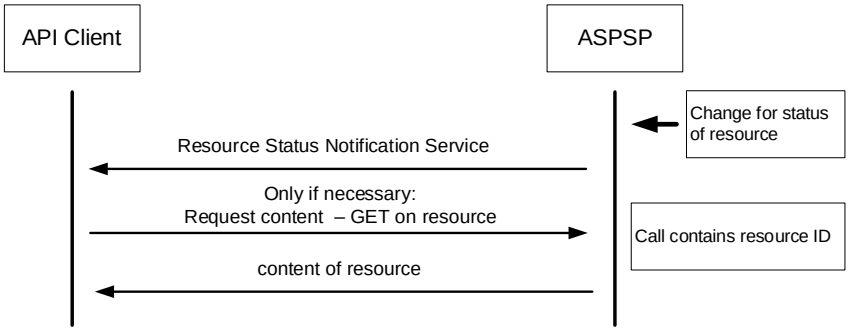


Figure 9: API Service RTP Status with Resource Status Notification Service

(ref. License Notice for full license conditions)

Remarks:

- The message of the Resource Status Notification function will indicate the change of the status of the resource, possibly including some parameters of the new status.
- The message of the lean Resource Status Notification function will also indicate if a following GET on the resource will be necessary to get further details about the new status.

Remark:

As already indicated before this API service may also be used

- by an Originator-RTPSP to get the status of an RTP from an Originator-RTPSP-Aggregator, if the Aggregator supports the openFinance API with this service as its passive interface to its clients, or
- by an RTP-Originator to get the status of an RTP from an Originator-RTPSP, if the Originator-RTPSP supports the openFinance API with this service as its passive interface to its clients. In case where the RTP has been initiated per bulk, the status information will be provided on bulk and single RTP entry level. For the latter, the RTP will be addressable as an own sub resource of the bulk resource.

5.2 RTP enhanced services for passive interface

The RTP enhanced services will be defined by a future version of this document. These RTP enhanced services will be the topic of a future market consultation.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

6 Example: Complete flow for RTP and corresponding payment

The following figure shows an example for a complete process starting with the initiation of an RTP by the API Client (Originator-RTPSP or Originator-RTPSP-Aggregator) for the major scenario. The process includes the initiation of the corresponding payment and the necessary SCA of the PSU. The propagation of this process to the RTP-Originator/Originator-RTPSP respectively the Originator-RTPSP/Originator-RTPSP-Aggregator interface is analogous.

Precondition:

> The consent of the PSU to receive an RTP from the RTP-Originator sent by the Originator-RTPSP (consent type "RTP-Originator Consent") or to receive an RTP from the Originator-RTPSP regardless of the RTP-Originator (consent type "Originator-RTPSP Consent") has been established before. This process to establish the necessary consent is not shown in the figure. This consent may have been established using the openFinance API of the ASPSP (as shown in section 5.1.1), by internal processes specific for the ASPSP or just between the PSU, the RTP-Originator and the Originator-RTPSP.

For the ASPSP two interfaces are shown: The interface to the API client. This is the openFinance API of the ASPSP supporting the RTP Services. The interface to the PSU can be any online channel used by the ASPSP for the communication with his account holder. This can be for example the online banking service of the ASPSP. This can also be the openFinance API of the ASPSP if the ASPSP supports direct access at his openFinance API.

In the figure it is assumed that the Resource Status Notification function is used.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

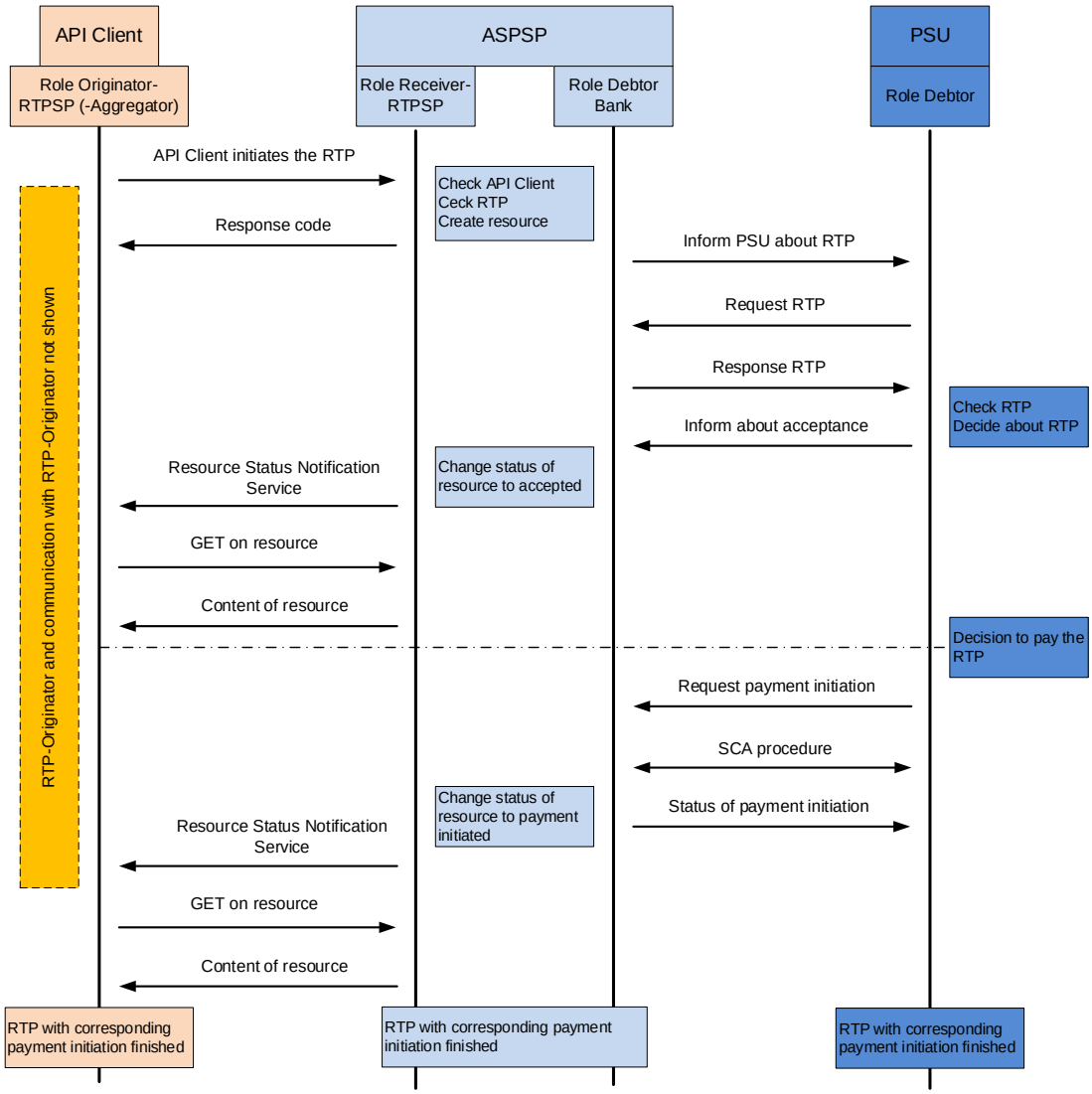


Figure 10: Example for a complete process with corresponding payment initiation

(ref. License Notice for full license conditions)

7 Key concepts of the openFinance API for the Request to Pay Service

This section contains an overview of key concepts of the openFinance API specific to the request to Pay Services. For basic key concepts please refer to [oFA-IG-ProtSec]. For the detailed specification of this extended service please refer to the document [oFA-IG-RtP].

7.1 Confirmation of the consent of the PSU (towards the API Client)

Transaction	How the PSU grants consent	How consent of the PSU is verified
Establishment of RTP consent	Identifies itself as part of the transaction, if necessary by strong customer authentication.	Verification of the identity of the PSU, if necessary by strong customer authentication.
RTP initiation	Consent was granted during a previously executed Establishment RTP consent transaction, or the consent has been confirmed/ established by some internal processes of the ASPSP as Receiver-RTPSP or internally between PSU, RTP-Originator and Originator-RTPSP.	The consent is not verified explicitly on API level, but might be implicitly verified, since an earlier API based consent mechanism might have impact on risk management tools of the ASPSP when doing fraud checks on incoming RTPs.
RTP cancellation	No consent of the PSU needed.	No consent to be verified.
RTP status	No consent of the PSU needed.	No consent to be verified.

Table 8: Consent of the PSU within RTP transactions

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

8 Operational rules

This section summarises the operational rules specifically to be observed by each API Client accessing the openFinance API for the Request to Pay Service and each ASPSP providing the Request to Pay Service at the openFinance API. In addition, the general operational rules as specified in [XS2A-OR-Core] apply.

Not all of these rules are enforced by technical means of the Extended Service within the openFinance API.

The order of the rules does not represent an order of importance.

8.1 API-Client authorisation

ASPSPs might refuse access to the openFinance API even if the related eIDAS certificate is valid, if not all related contract conditions are fulfilled. This might be controlled e.g. via a scheme directory.

8.2 API-Client identification

Accessing clients are identified based on eIDAS certificates (QWAC) in analogy to the Core XS2A Interface, but without mandating PSD2 specific attributes in the certificate. These certificates are only used for the identification of the API client. In general these certificates will not contain any scheme or service specific data.

TPPs according to PSD2 regulation already owning PSD2 compliant eIDAS certificates can reuse these certificates by their access to the openFinance API of an ASPSP.

8.3 Coding of business data

The API data coding of RTP data for XRTP-B will be JSON encoding. ASPSPs may support XML based data encoding in addition.

8.4 Payment initiation corresponding to an RTP

An ASPSP may restrict his Request to Pay Services such that the PSU as receiver of an RTP is committed to use this ASPSP also for the execution of payments corresponding to the RTP.

8.5 Payment initiation corresponding to an RTP via a PISP function

At the request of a PSU of an ASPSP as receiver of an RTP the corresponding payment may be initiated via the ASPSP itself or also by a PISP function of the Receiver-RTPSP through an account of the PSU held by another ASPSP. In any case the RTP transaction and the payment transaction will be seen as two different transactions based on two different resources. Nevertheless, a change of the status of the corresponding payment still enables the Receiver-RTPSP to mirror the payment status within the RTP transaction.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Remark: The payment status cannot be mirrored in cases, where the PSU is using third channels to execute the related payment.

8.6 RTP Cancellation

An API Client may cancel an RTP which he has submitted to the ASPSP before.

An RTP shall not be cancelled if a corresponding payment has already been initiated.

Further restrictions for the cancellation of an RTP may be defined by a scheme regulating the Request to Pay Services offered by an ASPSP.

8.7 Identification of RTP-Originator

The RTP-Originator has to be identified within an RTP or RTP consent using "creditor account" and "creditor id" and "creditor name".

It is the responsibility of the Originator-RTPSP transmitting the RTP (potentially via an aggregator) to an openFinance API that the identification of the RTP-Originator is correct.

8.8 Identification of the Originator-RTPSP

If the Originator-RTPSP is submitting the RTP as API Client the Originator-RTPSP is identified by the content of his QWAC.

If the RTP is submitted by an Originator-RTPSP-Aggregator as API Client, the Originator-RTPSP has to be identified within the RTP.

The ASPSP as Receiver-RTPSP may mandate special naming schemes e.g. LEI/BIC/PSD2 regulated License Number to identify the Originator-RTPSP.

In this case it is the responsibility of the Originator-RTPSP-Aggregator submitting the RTP or related RTP consents as API Client to an openFinance API that the identification of the Originator-RTPSP is correct.

8.9 Status of an RTP transaction

After receiving an RTP message the transaction status will have different values depending on the process of the RTP and the corresponding payments. The following values for the status are distinguished:

Rejected

The RTP has been rejected by the ASPSP directly for different reasons:

- API Client not known or not authorised to use the RTP Service of the ASPSP.
- Receiver of the RTP (PSU) not known by the ASPSP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Necessary consent of the PSU has not been established before.
- RTP technically not correct.

Or the RTP has been rejected by the PSU after the RTP has been presented to him.

Codes indicating the reason for rejection will be given with the response.

Pending

RTP technically accepted, but no checks about the PSU (as receiver of the RTP) have been done yet.

Received

RTP technically accepted, PSU is accepting RTP in general and is reachable by the ASPSP. RTP is available for PSU.

Presented

RTP has been presented to the PSU, but no answer from PSU yet.

PSU has had the opportunity to read the RTP (for example after log in to his online banking account).

Accepted by PSU

RTP has been presented to the PSU, PSU has accepted the RTP without any changes. Corresponding payments have not been initiated yet.

Accepted by PSU with Changes

RTP has been presented to the PSU, PSU has accepted the RTP but with changes. Corresponding payments have not been initiated yet.

Partially Accepted by PSU

RTP has been presented to the PSU, RTP contains different sub-articles, PSU has accepted some, but not all. Corresponding payments have not been initiated yet.

Payment Authorised

RTP has been accepted by PSU. Corresponding payment has been initiated, including the SCA of the PSU.

Payment Authorised after Changes

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

RTP has been accepted by PSU, but with some changes. Corresponding payment has been initiated, including the SCA of the PSU.

Payment Authorised after Partial Acceptance

RTP has been partially accepted by PSU. Corresponding payment has been initiated, including the SCA of the PSU.

Payment Partially Authorised

RTP has been accepted by PSU. Corresponding payment has been initiated. The payment has been authorised (with SCA) by at least one PSU, but necessary authorisation by further PSU is missing.

Payment Authorised after Changes

RTP has been accepted by PSU, but with some changes. Corresponding payment has been initiated. The payment has been authorised (with SCA) by at least one PSU, but necessary authorisation by further PSU is missing.

Payment Authorised after Partial Acceptance

RTP has been partially accepted by PSU. Corresponding payment has been initiated. The payment has been authorised (with SCA) by at least one PSU, but necessary authorisation by further PSU is missing.

Payment Completed

The corresponding payment initiated after an RTP has been completed.

Cancelled

RTP has been cancelled successfully.

Remarks:

- These values for the status of an RTP transaction will be mapped to ISO codes by the specification (see [oFA-IG-RtP].
    - This mapping may depend on the payment scheme used for the execution of the corresponding payments.
    - For the status of payments the same ISO codes will be used as they are used for the extended payment services.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- For a given transaction some of the values for the status may not be relevant or skipped. For example it is possible, that a transaction has got the status "Presented", and the next status value is "Payment Authorised", if the acceptance of the RTP and the initiation of the corresponding payment is done by the PSU in one step.

- The payment related statuses are not mandated to be provided by the ASPSP by this specification. This specifically applies to transaction statuses which are related not to the initiation, but the execution of the payment. Nevertheless, the Receiver-RTPSP should provide this information when available.

- The API client can always read the status by a GET message on the resource, which will be created by the ASPSP as result of the submission of the RTP with the request to the openFinance API of the API Client.

- For RTPs submitted as a bulk to an ASPSP in the role of the Originator-RTPSP, resources are created for every single RTP. The status of every RTP entry then can be retrieved by reading the status of the related entry resource.

- The bulk as such will only support restricted status values, i.e. Rejected, Pending, Received and Cancelled. Anyhow, it will transport the status of all entries received.

- If the API Client supports the Resource Status Notification Service, the ASPSP will inform the API Client about any change of the status of the resource by using this Resource Status Notification Service. A subscription to the Resource Status Notification Service for an RTP Bulk will lead to the ASPSP informing the API Client about every RTP entry status change. All these status changes will be posted to the same URI provided by the API Client during the implicit subscription during submission of the RTP.

8.10 Status Notification

If the ASPSP uses the Resource Status Notification Service to inform the API Client about a change of the status, the ASPSP will start three attempts to push the status information to the URI given by the API Client as part of the RTP initiation request.

If the third attempt is not successful the ASPSP "stays silent" for this change of the status. Of course information about the new status will still be available for the API Client using the GET access method.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

9 Message and data model

In the following, an abstract data model is presented for the specific usage of the Request to Pay Service within the openFinance API. The basic abstract data model for the openFinance API is defined in [XS2A-OR-Core].

A detailed data model for this Request to Pay Service is defined in [oFA-IG-RtP].

9.1 Protocol Level

There are no specific requirements on data modelling on protocol level. The following data elements are used independently of the semantic of the related messages, building an abstract basic protocol level.

9.2 Authorisation Data Model

Not relevant for the Request to Pay Services.

9.3 Request to Pay related data model

Within the openFinance API, an RTP transaction is initiated by a pair

- RTP Initiation Request and RTP Initiation Response.

After the RTP transaction has been initiated the API Client can only change the status of the RTP transaction by cancelling the RTP submitted before. This is done by a pair

- RTP Cancellation Request and RTP Cancellation Response.

For all other changes of the status of an RTP transaction the API Client (in its role as Originator-RTPSP or Originator-RTPSP-Aggregator) is not involved directly.

An API Client may request information about the status of an RTP transaction at any time. This is done by a pair

- RTP GET Status Request and RTP GET Status Response.

This information request and response are implemented by using the usual http GET method on the dedicated status attributes of the resource which has been created by the ASPSP as result of the RTP Initiation Request. Of course the complete content of the resource can also be retrieved by the related GET access method.

Remark:

Data elements are marked mandatory, conditional or optional as seen from the definition of the openFinance API. These rules may be strengthened if necessary by a scheme regulating the Request to Pay Services offered by an ASPSP, i.e. a scheme

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

may mark a data set as mandatory/conditional if it is marked optional in the following lists.

9.3.1 RTP Initiation Request

In the following, a minimum set of requirements on the Payment Submission Request is defined. These requirements are independent of the encoding.

9.3.1.1 PSU data

- None for an RTP Initiation Request.

9.3.1.2 API Client relevant data

- API-Contract-ID (mandatory)

  The API Client needs to add the Contract Identification, which refers to the service contract agreed on during onboarding to the ASPSP.

- Notification-URI (optional)

  URI of the API Client for the Resource Status Notification Service. The ASPSP shall send status information to this URI if the Status Notification Service is supported by the ASPSP. Only to be provided if the API Client uses the Resource Status Notification Service.

- Notification Content Preference (optional)

  This data element defines the preference of the API Client to receive all or only the final status change.

- Brand Logging Information (optional)

  This header might be used to inform the ASPSP about the brand used by the API Client.

- RTP Scheme Identifier (mandatory)

  Identification code of the scheme (mandatory) transported explicitly or implicitly already as part of the contract-id parameter of the API Client.

Note: API Client parameters of this section marked as optional may be ignored by the ASPSP, if not agreed otherwise in service contracts..

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

9.3.1.3 RTP Data

Payment Related Data

This is data needed by the ASPSP (Receiver-RTPSP) to generate the requested payment to be presented to the PSU for authorisation in the related ASPSP online channel. Note that the RTP data modelling is a full integration into the payment data model of the openFinance API Framework, i.e. the payment identifiers for example contained will at the same time be used as RTP identifiers. Details are described in the document [oFA-IG-RtP]. Please note that the party descriptions for the creditor and the ultimate creditor as well as for debtor and ultimate debtor are supporting not only name, address and identification fields but also trade names and merchant category codes. The same elements will also be supported for Extended Payment Initiation Services, cp. [oFA-OR-EPIS].

The RTP-Originator has to be identified uniquely within an RTP using "creditor account" and "creditor id" and "creditor name". See also section 8.7.

Request Related Data

- Identifier of the Originator-RTPSP Data (conditional)

  This identifier

  - will be transported in the eIDAS certificate if the API Client equals the Originator-RTPSP,
  - will be transported in the request related information if the API Client equals the Originator-RTPSP-Aggregator,
  - is transported as part of the URI if the API Client equals the RTP-Originator.

- Identifier of the Receiver-RTPSP (mandatory)

  This identifier

  - will be transported in the URI if the API Server equals the Receiver-RTPSP
  - will be transported in the request related information if the API Server equals the Originator-RTPSP-Aggregator or the Originator-RTPSP.

- Further RTP Data sent in the request related information

  - Expiry Date/Time of the RTP (mandatory)
  - Attachment sent by creditor to debtor (optional) (supported only by future versions)
  - Date and time stamp of the RTP (mandatory)
  - Remittance information for the debtor (optional)

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Creditor's payment conditions (optional)

Document related data

For the document container the ISO data type Document12 as defined in section 4.1.9.3 of [ISO-CPAR] will be used to transport the (binary) document and related meta data. More details are provided in [oFA-IG-RtP].

9.3.2 RTP Initiation Response

The response to an RTP Initiation Request will contain only the following data:

- Initiation ID

  A unique identification provided for this RTP initiation.

- Transaction status

  Status of the Transaction after the initiation of the RTP. See section 8.9 for possible values. The status will be given by two different attributes for the payment transaction and the request transaction. Possible values for the payment status are as for the usual payment initiation at the API.

- Extended status information

  The following additional attributes are provided as extended status information:

  - Status reason (additional information about the reason for example for the rejection of a request)

- Charges related information (formatted as for payment initiation)

- Status notification confirmation

  By this parameter the ASPSP indicates if he will inform about the status changes using the Resource Status Notification Service.

- Status notification content

  By this parameter the ASPSP indicates if he will inform about all or only the final status change using the Resource Status Notification Service.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

9.3.3 RTP Cancellation Request

RTP to be cancelled has to be identified by the URI (location of the resource) and the Initiation ID.

Cancellation is performed by the DELETE access method.

9.3.4 RTP Cancellation Response

Success or denial of the cancellation request is transported by the related standard http response code.

9.3.5 RTP GET Status Request

Done by the GET access method on the resource identified by the URI (location of the resource) and the Initiation ID.

9.3.6 RTP GET Status Response

- Transaction status

  Status of the Transaction after the initiation of the RTP. See section 8.9 for possible values. The status will be given by two different attributes for the payment transaction and the request transaction. Possible values for the payment status are as for the usual payment initiation at the API.

- Extended status information

  The following additional attributes are provided as extended status information:

  - Status reason (additional information about the reason for example for the rejection of a request)
  - Debtor decision date/time (timestamp of the acceptance/rejection of the RTP by the PSU)
  - Accepted amount (contained only in case of deviation from the instructed amount)
  - Acceptance date/time (contained only if the agreed execution date/time deviates from the requested execution date/time)
  - Accepted payment instrument (payment instrument used by the PSU for the related payment)
  - Status identification (reference of the PSU to the confirmation process if provided)

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

10 Annex

10.1 Glossary

AIS

Account Information Service according to article 4 (16) of [PSD2] and as regulated by article 67 of [PSD2].

AISP

Payment service provider offering an AIS to its customer. See article 4 (19) of [PSD2].

ASPSP

Account Servicing Payment Service Provider providing and maintain a payment account for a payer. See article 4 (17) of [PSD2].

Onboarding

In this document the term onboarding is only used for the (technical) onboarding of an API client at the openFinance API of an ASPSP according to [oFA-OR-ADM].

PIISP

Payment Instrument Issuer Service Provider according to article 4 (14) and 45) of [PSD2]. A PIISP can use the service "Confirmation on the availability of funds" as regulated by article 65 of [PSD2].

PIS

Payment Initiation Service according to article 4 (15) of [PSD2] and as regulated by article 66 of [PSD2].

PISP

Payment service provider offering a PIS to its customer. See article 4 (18) of [PSD2].

PSP

Payment service provider according to article 4 (11) of [PSD2].

PSU

Payment Service User according to article 4 (10) of [PSD2].

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

QTSP

Qualified Trust Service Provider, e. g. a trust centre issuing qualified certificates-

QWAC

Qualified certificate for website authentication issued by a QTSP.

RTP

Request to Pay

RTPSP

Request to Pay Service Provider

SCA

Strong Customer Authentication – authentication procedure based on two factors compliant with the requirements of [PSD2] and [RTS].

TPP

Third Party Provider – generic term for AISP/PIISP/PISP.

TSP/QTSP

Trust Service Provider according to [eIDAS]. Within the context of the XS2A interface specification only qualified TSPs (QTSPs) according to section 3 of [eIDAS] issuing qualified certificates for electronic seals and/or qualified certificates for website authentication which are compliant with the requirements of [RTS] are relevant.

XS2A interface

Access to account interface – interface provided by an ASPSP to TPP for accessing accounts.

10.2 References

[XS2A-IG-Core] NextGenPSD2 XS2A Framework, Implementation Guidelines, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3.12, published 01 July 2022

[XS2A-OR-Core] NextgenPSD2 XS2A Framework, Operational Rules, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3, published December 2018

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

[oFA-IG-Consent] openFinance API Framework, Implementation Guidelines, Consent API for V2.x, Version 2.0, 05 October 2023

[oFA-IG-ProtSec] openFinance API Framework, Implementation Guidelines, Protocol Functions and Security Measures, Version 2.0, 05 October 2023

[oFA-OR-PAIS] openFinance API Framework, Operational Rules, Push Account Information Services, Version 1.1, 3 March 2022

[oFA-IG-RtP] openFinance API Framework, Implementation Guidelines for Extended Services, Request to Pay Services, version 2.0, 20 October 2023

[oFA-OR-ADM] openFinance API Framework, Operational Rules for Extended Services, Administrative Services, draft version 0.9, 27.05.2021

[oFA-OR-EPIS] openFinance API Framework, Operational Rules, Extended Payment Initiation Services, Version 1.0,. 29 April 2022

[eIDAS] Regulation (EU) No 910/2014 of the European Parliament and of the Council on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market, 23 July 2014, published 28 August 2014

[ISO-CPAR] ISO 20022, Creditor Payment Activation Request – Maintenance 2020 – 2021, Messaged Definition Report -Part 2, Approved by the Payment SEG on 11 January 2021

[PSD2] Directive (EU) 2015/2366 of the European Parliament and of the Council on Payment Services in the Internal Market, published 25 November 2016

[RTS] Commission Delegated Regulation (EU) 2018/§() of 27 November 2017 supplementing Directive 2015/2366 of the European Parliament and of the Council with regard to Regulatory Technical Standards for Strong Customer Authentication and Common and Secure Open Standards of Communication, L69/23, Official Journal of the European Union, 13.03.2018

[TS 119 495] Draft ETSI TS 119 495, Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificates Profiles and TSP Policy Requirements under the Payment Service Directive 2015/2366/EU, V1.1.2 (2018-07)

10.3 List of figures

Figure 1: Core XS2A interface and openFinance API 2

Figure 2: Basic model of Request to Pay 6

Figure 3: Distinction Payment Layer ↔ Request to Pay Layer 9

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Figure 4: Architecture for the first step of the basic RTP Services of the openFinance API Framework 12
Figure 5: API Service Establishment of RTP consent 25
Figure 6: API Service RTP Initiation 26
Figure 7: API Service RTP Cancellation 27
Figure 8: API Service RTP Status with polling 28
Figure 9: API Service RTP Status with Resource Status Notification Service 28
Figure 10: Example for a complete process with corresponding payment initiation 31

10.4 List of tables

Table 1: Mapping of RTP actors to roles of the openFinance API Framework 10
Table 2: Scenarios considered for API Client 11
Table 3: Mapping of RTP actors to roles of the openFinance API Framework 14
Table 4: Mapping of RTP actors to roles of the openFinance API Framework 15
Table 5: Request to Pay Services scheduled for step 1 of further detailed work 20
Table 6: Request to Pay Services scheduled for step 2 of further detailed work 21
Table 7: API services of the RTP basic services for passive interface 23
Table 8: Consent of the PSU within RTP transactions 32

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)