DESP Experimentation Portal - User Guide

This document is intended solely for the purposes of the Pioneer Workstream exercise and does not represent any aspect of a potential real-world digital euro implementation.

Introduction to the project

The Pioneer Workstream originates from the broader initiative of Innovation Partnerships, a project promoted by the European Central Bank (ECB). This initiative was launched in late October 2024, when the ECB issued a call for interest, featured on its website. The aim was to attract two categories of participants, namely pioneers and visionaries, eager to collaborate and contribute to the development of use cases on conditional payments and innovative ideas for digital euro. This document focuses on the Pioneer Workstream.

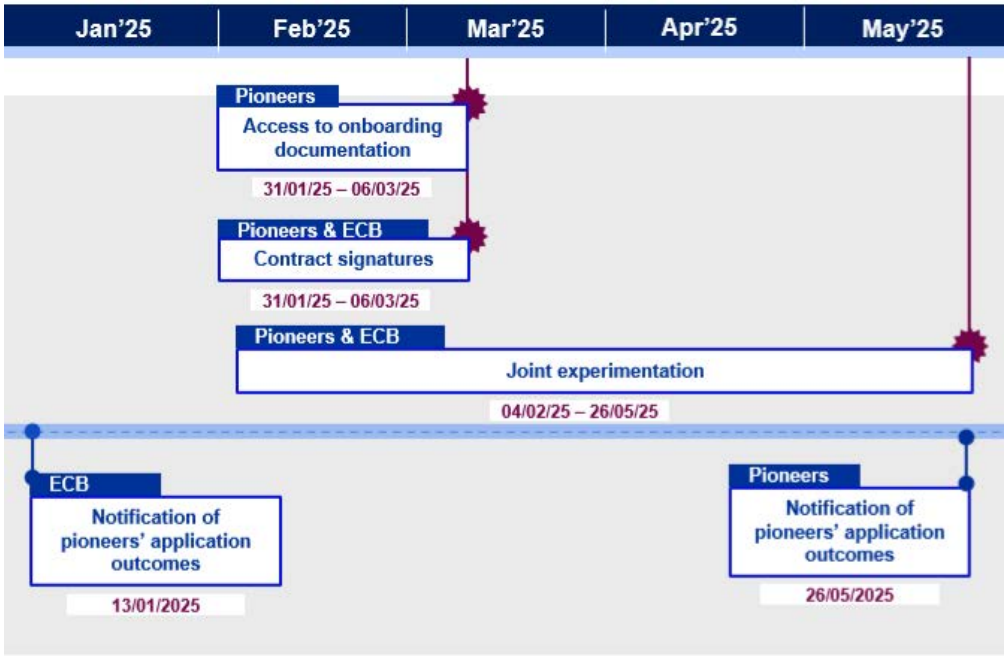
Conditional Payments

Conditional payments refer to transactions that are automatically executed when predefined conditions are met, offering a new level of automation and security in digital transactions. Within the context of the ECB digital euro initiative, conditional payments are being explored for their functionality as a reservation of funds.

This mechanism can automate payment scenarios, reducing the need for manual oversight and minimising transaction errors, thus enhancing overall efficiency. The use cases for conditional payments are extensive, and the goal of this initiative is to engage with a variety of stakeholders to gather insights into the practical applications and technical feasibility of conditional payment use cases.

Timeline

The project follows the following tentative timeline:



We welcome your feedback on these updates—your insights are vital to the continued improvement of the platform and the ongoing development of the digital euro project. Thank you for being an essential part of this journey.

Roadmap/Future Release

The current implementation of digital signature validation in REST API v3 has some limitations. For more details, please refer to the **Technical Details** section.

We sincerely appreciate all the effort invested throughout—from implementing use cases to actively participating in feedback rounds. Your contributions have been invaluable in shaping and advancing this initiative.

Technical Details

The Digital Euro Service Platform (DESP) API version 1 is a RESTful API designed to facilitate experimentation with the ECB's digital euro initiative for 2024/2025. Tailored for conditional payment flows, the DESP API enables stakeholders to prototype and validate use cases for digital euro adoption in a secure and isolated ecosystem.

The API enables a single pioneer to act as either the PSP for the payer, the PSP for the payee, or both, in controlled scenarios. This setup is designed to simulate real-world transactions between two different PSPs without requiring cross-PSP coordination during testing. By allowing a single pioneer to handle both the payer and payee PSP roles, the experimentation focuses on validating functionality in a simplified and controlled environment, as well as communication with the ECB’s digital euro service platform (DESP).

API Authentication

The API currently uses BasicAuth for authentication. Please use the same username and password that you use to access this portal.

Versioning

The API follows semantic versioning. Non-breaking changes include the addition of new REST endpoints, new fields in response payloads, and non-mandatory fields in request payloads, so please configure your client to accommodate these enhancements.

- **rest-api-1 (1.4.0)**
- **rest-api-2 (2.0.13)**
- **rest-api-3 (3.0.3)**
- **grpc-api-1 (1.0.5)**

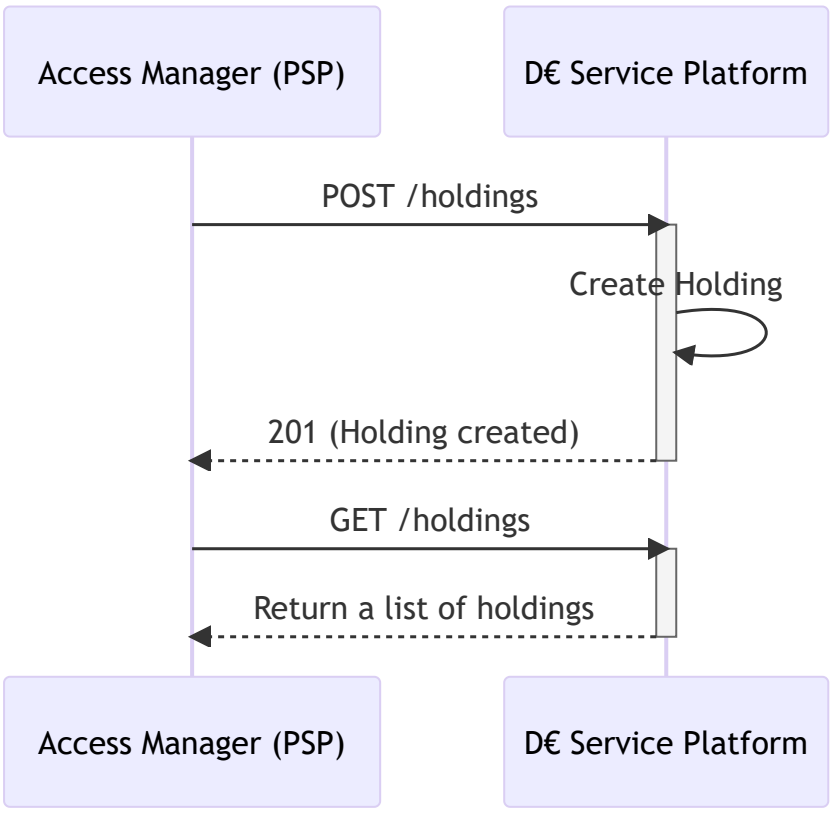
Application Live Information

Service Info

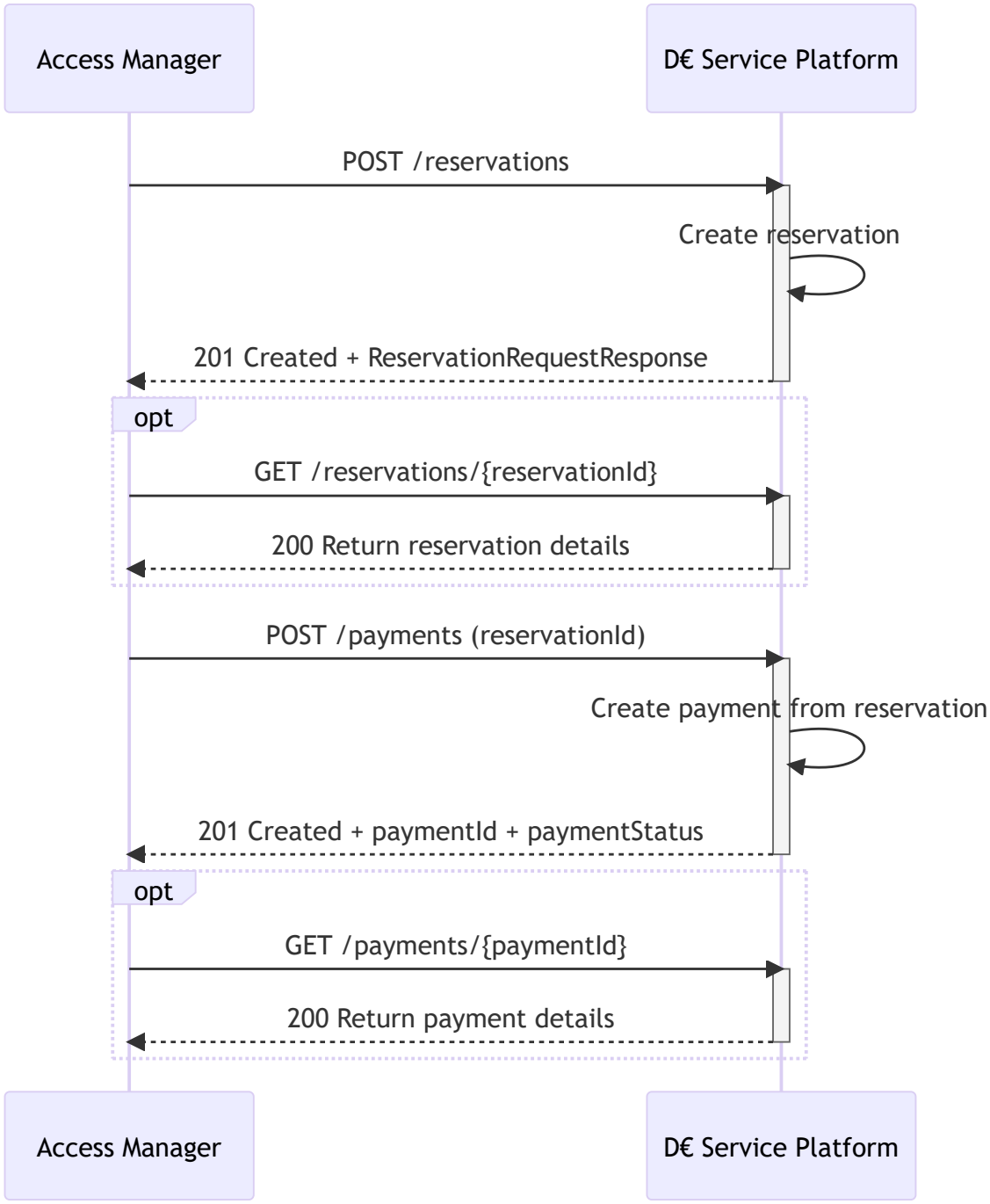
Service Name: desp-experimentation Service
Version: 3.0.2
Start Time UTC: 2025-06-26 08:51
Build Time UTC: 2025-06-26 08:32
Node: java-instance-1

Workflows (API version 1)

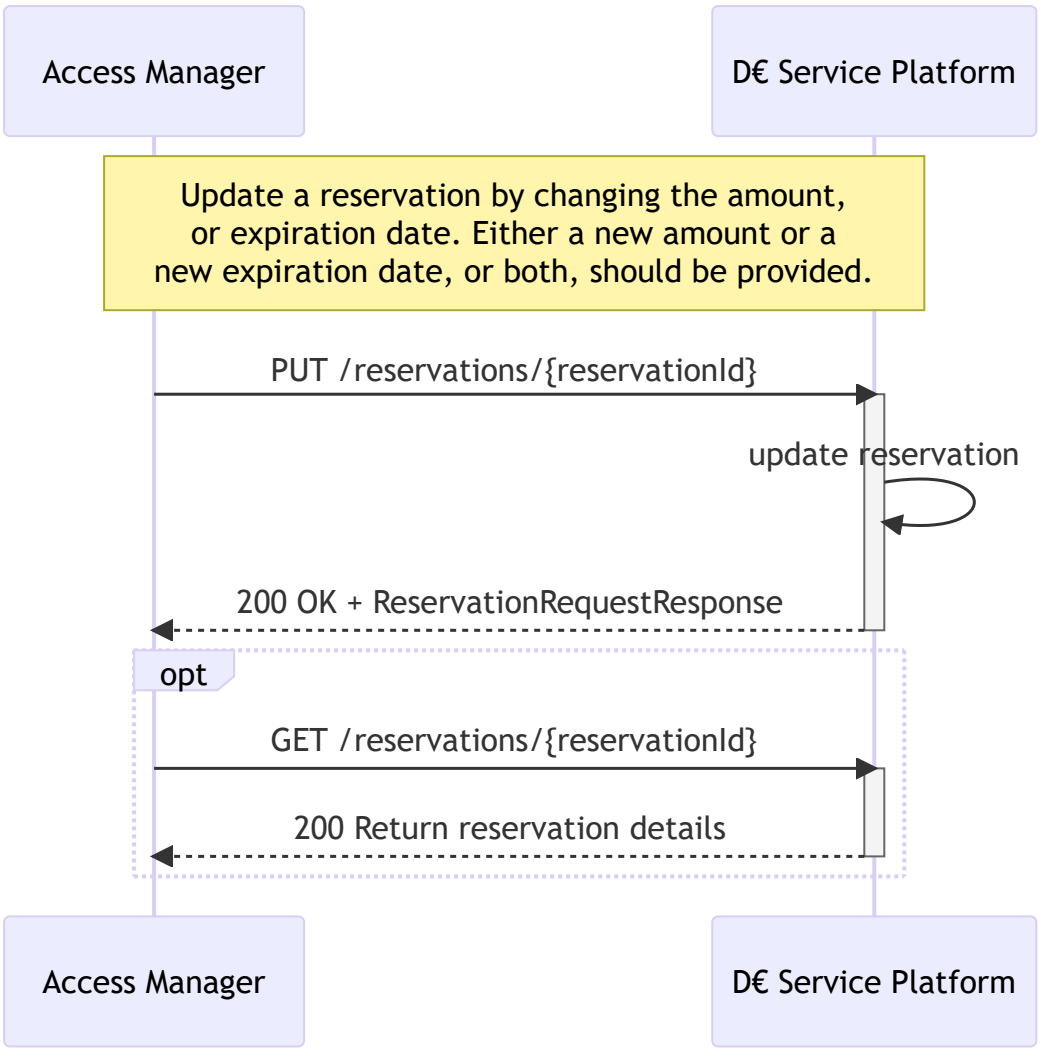
Holdings creation and retrieval



Reservation and payment creation and retrieval

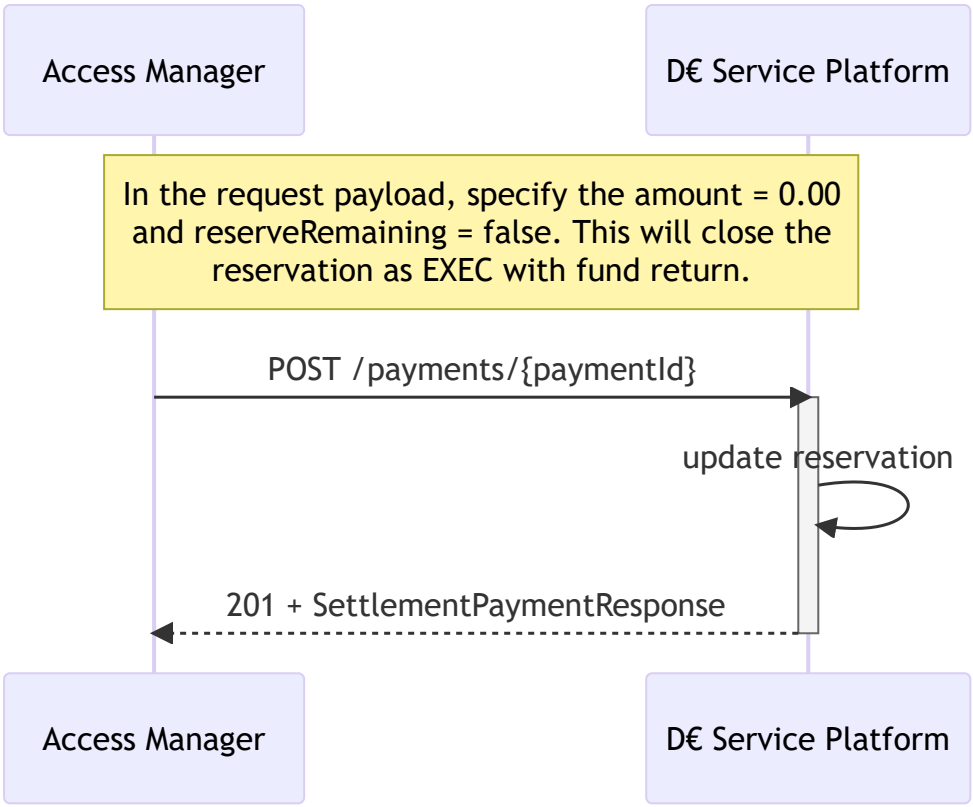


Reservation update



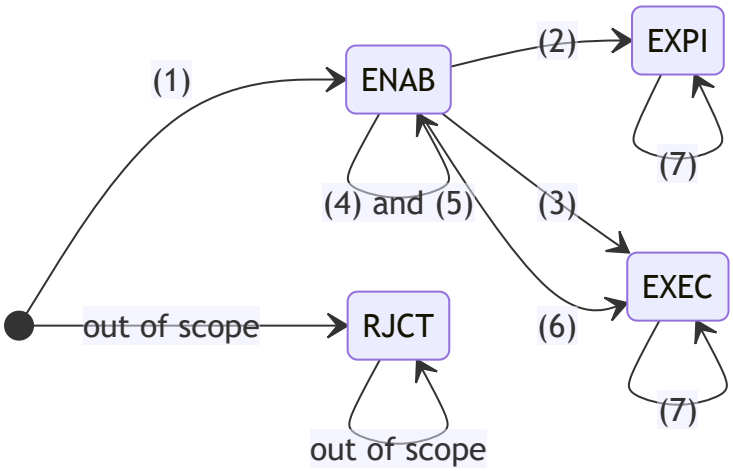
Reservation cancellation

In the experimentation, the cancellation of a reservation is done by sending a POST request to the payment endpoint with the following parameters: `amount = 0.00` and `reserveRemaining = false`. This will close the reservation as EXEC with fund return.



Reservation state transition diagram

Below is the state transition diagram for the reservation lifecycle. The diagram illustrates the various states a reservation can be in, along with the events that trigger transitions between these states.



Callout Explanations:

- (1) **Creation:** Marks the initial state when a reservation is created.
- (2) **Expiry:** Indicates the transition to `EXPI` when the expiry date passes without full payment.
- (3) **Full Payment:** Shows the change to `EXEC` when a payment fully settles the reserved amount.
- (4) **Partial Payment with Reserve:** Covers cases where a partial payment with `reserveRemaining=true` keeps the reservation open.

- (5) **Overpayment Rejection:** Applies when a payment exceeds the reserved amount, maintaining the `ENAB` state.
- (6) **Reservation Cancellation:** Occurs when a payment with `reserveRemaining=false` and `amount = 0.00` closes the reservation as `EXEC` with fund return.
- (7) **Payment Not Possible:** If the status of a reservation is `EXEC` and `EXPI`, no further payments on this reservation would be possible.

gRPC notes

Below we would like to share some information that might be helpful when testing the gRPC API. We are neutral on the tools you use, do not encourage any specific tools, and do not offer any support around them. Also, you are the only responsible for the tools you use.

When integrating or testing the gRPC API please consider the network restrictions your company might enforce (e.g. HTTP proxies, firewalls).

If you would like to try the gRPC API in HTTPYac, here is the code snippet as a starting point (replace `<username>` and `<password>`):

```
proto < ../../desp-api/grpc/desp.proto


### Query Info gRPC endpoint
GRPC desp.dev/api.grpc.v1.Desp/GetInfo
Content-Type: application/grpc
Authorization: Basic <username> <password>
ChannelCredentials: SSL

{}
```

If you would like to try grpcurl, here is a command:

1. Concatenate the username and password into a single string: `<username>:<password>`.
2. Encode the concatenated string using Base64 encoding.
3. Please add the newly obtained `<Base64-encoded>` credentials below.
4. Remember to download the gRPC proto file by clicking on `Download spec` link in Versioning section.
5. Save the proto file `desp.proto` in the same folder where `grpcurl` is located.

```
grpcurl -proto desp.proto -H "Authorization: Basic <Base64-encoded>" desp.dev:443 api.grpc.v1.Des
```

Dedicated Cash Account (DCA) Information

DCA Information

- **Account ID:** 88852ed4-6cb5-43c3-9760-2395feda4903
- **Balance:** 10000.00 €

Context of the DCA and technical information

The DCA feature enables both the reverse waterfall and waterfall mechanisms in the context of conditional payments. It allows you to combine funding and defunding functionalities when reserving and making a payment, respectively, as specified in the API v2 under the technical details section.

The `DCA Account ID` must be included in the payload request whenever a funding or defunding object is present.

The `Balance` provides the real-time amount available in your Dedicated Cash Account (DCA). Please note that we have created a DCA for each of you, and it will be used — with its balance deducted or increased — when funding and defunding, respectively. The `Balance` reflects the amount of money you, as a PSP, have at any given moment. You will need to refresh this page to obtain the most up-to-date Balance information.

Release Notes / Change Log

2025-06-25 - V3.0.1

This new release provides the following updates:

V3 REST API:

- Implementation of `GetPayment()` with basic details related to different types of payment.

V2 REST API:

- Implementation of `GetPayment()` with basic details related to different types of payment.

2025-06-18 - V3.0.0

This service release implements the REST-API v3 specification with support for digital signatures. Please mind that there are some limitations and some of them will be fixed in future releases. Check the "**Technical Details**" page for information.

REST API versions v1 and v2 do not support digital signatures.

2025-06-03 - V2.3.0

This new release provides the following updates:

V2 REST API:

- Enhanced reservation to allow funding via reverse waterfall mechanisms.
- Enhanced payment endpoint with:
  - **P2M (Person-to-Merchant, via reservation)**: Includes a defunding object to enable a waterfall mechanism when the merchant surpasses the holding limit of 0. This is a mandatory object in V2 REST API.
  - **M2P (Merchant-to-Person, Refunds)**: Includes a funding object to allow reverse waterfall mechanisms, enabling merchants to process payments.
  - **P2P (Person-to-Person)**: Supports both funding and defunding objects as optional parameters.

V1 REST API:

The V1 version will remain available but will not include the enhanced functionality described above.

2025-05-22 - V2.2.3

This new release provides the following updates:

V2 REST API:

- Server-generated UUIDs for holding, reservation, and payments.
- Enhanced reservation creation with the addition of a funding object, allowing you to create reservations exceeding the available balance by funding them through DCA. **(Funding via Reverse waterfall)**
- Every pioneer will have access to an encoded Available DCA within the **DCA Information** section of the portal, which includes:
  - **UUID Entry**
  - **DCA Amount**: This amount will be reset to €10,000 every night.

V1 REST API:

The V1 version will remain available but will not include the DCA functionality.

2025-04-22 - V2.2.0

This service version implements API version 1.4 with support for querying reservations with the following optional query parameters: `createdFrom`, `createdTo`, `reservationStatus`, `debtorEntry`, and `creditorEntry`.

2025-04-11 - V2.1.0

The following gRPC services are now additionally exposed, covering the functionalities of REST-API v1.3.0:

- **Reservation:** `CreateReservation()`, `UpdateReservation()`, `GetReservation()`
- **Payment:** `CreatePayment()`, `GetPayment()`

2025-03-27 - V2.0.0

First version with gRPC support. Following gRPC services are exposed:

- **Desp:** `GetInfo()`

- **Holding:** `GetHoldings()`, `GetHolding()` and `CreateHoldings()`

2025-03-14 - V1.1.10

Various bug fixes and enhancements.

Bug Fixes

- Expired and executed reservations can now be retrieved.

2025-03-04 - V1.1.3

Summary

This service version implements API version 1.3. This update includes the following features:

- Supports operations for creating, modifying, and retrieving the status of reservations.
- Enables operations for processing payments related to reservations and retrieving the payment status.
- Supports the already existing endpoints, namely the creation and retrieval of holdings from creditors and debtors. Please see the Workflows for a visual representation.

2025-02-04 - V1.0.19

Summary

This service version (V1.0.19) implements the API version 1.1, that supports creation and retrieval of holdings.

Frequent Asked Questions

1. Is the mailbox the sole communication channel available at the moment?

Yes, currently, we are managing communication with the pioneers through a dedicated mailbox. However, we are open to scheduling individual meetings or calls if necessary to address specific communication needs. To assist us in evaluating cases, we encourage pioneers to provide a detailed description of their issues or inquiries via email, allowing us to direct requests to the appropriate team (IT or Business).

2. What is the onboarding package and what does it include?

The onboarding package is the portal page, a dynamic resource designed to provide pioneers with the necessary information for technical integration with the ECB Backend infrastructure. It serves as a comprehensive hub containing detailed documentation, guidelines, and tools to help pioneers quickly familiarise themselves with the system's functionalities and processes. By utilising the portal page, pioneers have all the essential information and support needed to ensure a successful partnership though the experimentation exercise.

4. Is the workflow shown on the portal page up to date?

The current workflows available on the portal page are updated based on the deployment of new endpoints. This will ensure that pioneers can understand the flow of information on a visual format, and thus helping in the integration efforts.

5. How do I provide feedback on the experimentation?

We intend to distribute a feedback form every three weeks to ensure alignment with expectations and streamline any requests from pioneers. Our goal is to ensure that pioneers' ideas are considered, thereby supporting the implementation of use cases and the testing of conditional payments.

6. When will the full API be available?

Following every new deployment, we will release the full API specifications, which will include all endpoints deployed thus far. This will ensure that pioneers have comprehensive access to all available features and functionalities. As we continue to develop and deploy additional endpoints, we will update the specifications accordingly to keep everything current and accurate.

7. What can we expect in future releases?

They will be aligned with the business scope, and the ‘Roadmap/Future release’ section will provide a summary of what will be included. Additionally, these updates will be informed by feedback from pioneers.

8. Are there examples available for the API?

Yes, the API documentation includes examples to help pioneers understand and implement its features effectively. These examples provide guidance on all the endpoints made available, making it easier for you to integrate them into their projects.

9. Will cross-PSP transactions be included in the scope of the experimentation? We are currently assessing your needs based on feedback. In future releases, cross-PSP and any other features might be included. We welcome any feedback or suggestions you may have to help us better meet your requirements.

10. Will there be additional endpoints and features?

Yes, endpoints and new features will be deployed and made available for pioneers to rely on them to test their use cases. In the future release section in the portal page, pioneers would be able to check which features will come out in the next major release. We strongly encourage pioneers to actively engage with us to provide feedback and offer suggestions for successfully working on use cases. Information on endpoints and new features will be included in the API specifications.

11. Which API interface will be utilised?

Currently, we are providing a REST API interface. In the next major release, we plan to introduce gRPC.

12. Will currency amounts be displayed using decimals? Currently, currency amounts are presented in string format. However, we are gathering feedback from pioneers to potentially display them in another data format.